감성분석을 위한 병렬적 HDFS와 맵리듀스 함수

백봉현*, 류윤규**

A Parallel HDFS and MapReduce Functions for Emotion Analysis

BongHyun Back, Yun-Kyoo Ryoo

요 약

최근 대량의 SNS(Social Network Service) 데이터로부터 유용한 정보를 추출하고 사용자의 진의 정보를 평가하기 위한 오피니언 마이닝(opinion mning)이 소개되고 있다. 오피니언 마이닝은 대량의 SNS 데이터로부터 빠른 기간 내에 데이터를 수집하고 분석하여 목적에 적합한 정보를 추출하는 효율적인 기법이 필요하다. SNS에서 발생되는 다양한 비정형 데이터로부터 감성정보를 추출하기 위해, 본 논문에서는 하둡(Hadoop) 시스템 기반의 병렬적 HDFS(Hadoop Distributed File System)와 맵리듀스(MapReduce) 기반 감성분석 함수를 제안한다. 실험결과로 제안한 시스템과 함수는 데이터 수집과 적재시간에 대해 O(n)보다 빠르게 처리하며, 메모리와 CPU 자원에 대해 안정적인 부하분산이 이루어지는 것을 확인하였다.

키워드: 빅데이터, HDFS, 맵리듀스, 감성분석, 비정형데이터분석

Abstract

Recently, opinion mining is introduced to extract useful information from SNS data and to evaluate the true intention of users. Opinion mining are required several efficient techniques to collect and analyze a large amount of SNS data and extract meaningful data from them. Therefore in this paper, we propose a parallel HDFS(Hadoop Distributed File System) and emotion functions based on Mapreduce to extract some emotional information of users from various unstructured big data on social networks. The experiment results have verified that the proposed system and functions perform faster than O(n) for data gathering time and loading time, and maintain stable load balancing for memory and CPU resources.

▸ Keyword : Big Data, HDFS, MapReduce, Emotion Analysis(감성분석), Unstructed data analysis(비정형데이터 분석)

• * 제1저자 : (주)아르고스 대표이사 **교신저자 : 대구보건대학교 보건의료전산과 교수
• 투고일 : 2014. 10. 30, 심사일 : 2014. 11. 30, 게재확정일 : 2014. 12. 30

Ⅰ. 서 론

최근 스마트폰 사용이 활성화되어 가는 사회적 분위기 속에서 대량의 SNS(Social Network Service) 데이터를 이용하여 의미있는 사용자 정보를 추출하는 오피니언 마이닝(Opinion Mining)이 대두되고 있다. 특히, 오피니언 마이닝은 SNS상에서의 사용자의 의도와 진의파악 등에 이용되어지고 있다. 이러한 오피니언 마이닝에서는 SNS 상에서 발생하는 대량의 데이터로부터 의미있는 정보를 신속하게 추출하고 이를 처리하는 기술이 필수적이다. 즉, SNS상의 데이터를 빠르게 분석하여 의미있는 정보를 추출하고, 이를 통해 대중들이 요구하는 의견과 생각들을 실시간으로 파악하여, 제품을 생산하고 서비스를 제공하는 다양한 분야에서 활용할 수 있도록 하는 기술이 필요하다. 또한 이러한 정제된 유효하고 다양한 정보들을 빅데이타 처리 분석기술을 통해 보다 효율적으로 관리하고 시각화하는 기술도 필요하다. 따라서, 본 연구에서는 SNS 상에서 발생되는 다양한 데이터, 특히 비정형 데이터로부터 사용자의 감성을 분석할 수 있는 하둡(Hadoop)기반의 병렬적 HDFS(Hadoop Distributed File System)와 맵리듀스(MapReduce) 기반의 감성분석 함수를 제안한다.

Ⅱ. 관련연구

정보 전달 대상 간의 상호 연결 방법이 용이하며, 데이터 작성 형식이 비교적 자유롭기 때문에 SNS상에서 발생하는 데이터는 대부분 비정형 데이터(unstructured data)이다. 비정형 데이터는 숫자 데이터와 달리 그림이나 영상, 문서처럼 형태와 구조가 복잡해 정형화되지 않은 데이터로 정의할 수 있다[1]. SNS 상에서 발생하는 수많은 비정형 데이터로부터 의미 있는 정보를 추출하기 위해서는 우선 비정형 데이터에 대한 처리가 필요하다.

비정형 데이터 분석은 형태소 분석을 기반으로 다양한 분석 방법[2-4]들이 연구되고 있다. 그러나 다양한 방송매체와 젊은 계층들로부터 새로운 유행어와 협의되지 않은 단어, 기호문자를 통한 기호단어, 단어들의 붙여 쓰기 등 데이터 분석을 저해하는 요소들이 발생하고 있다. 이에 따라 컴퓨터를 통한 언어 분석과 감성 분석이 어려워지고 있고 이에 대한 유효성 검정이 더욱 어려워지고 있다.

따라서, 비정형 또는 반정형 텍스트 데이터에서 자연언어처리 기술에 기반하여 정보를 추출하고 가공하는 텍스트 마이닝(Text mining)[5]에 관한 연구[6-7]가 진행되고 있다. 이들은 대량의 텍스트 데이터들로부터 의미 있는 정보를 추출하거나, 연관된 정보들을 정제하기 위한 사전기반, 기계학습기반의 통계적, 규칙적 알고리즘을 사용하고 있다. 또한 텍스트에서 긍정(Positive), 부정(Negative), 중립(Neutral)의 선호도를 판단하는 오피니언 마이닝(Opinion Mining)에 관한 연구도 진행되고 있다[8-9].

현재, 빅데이타 처리를 위한 다양한 오픈소스 프로젝트들을 하둡에코시스템(Hadoop ECO system)[10]으로 명명하여 진행하고 있다. 빅데이터 처리에 사용되는 데이터베이스는 전통적인 관계형 데이터베이스보다 덜 제한적인 일관성 모델을 이용하는 데이터 저장 및 검색을 위해 NoSQL(Not-Only SQL)[11]을 이용한다. NoSQL은 기존 RDBMS의 SQL과 같은 관계형 데이터베이스뿐만 아니라 상황에 맞는 데이터베이스를 사용한다. 즉, 기존의 관계형 데이터 베이스에서 볼 수 있는 수직적 확장성(scale up)에서 수평적 확장성(scale out) 형식을 제공하며, 테이블 스키마와 테이블간의 조인이 없으며, 읽기/쓰기에 빠른 응답시간과 확장성을 제공한다. 현재 업계 및 학계에서 NoSQL데이터베이스에 관한 많은 연구가 진행되고 있으며, 대표적으로 구글의 BigTable[12], 아마존의 Amazon DynamoDB[13], 오픈소스 프로젝트의 Apache HBase[14], Cassandra[10], MongoDB[15] 등이 대표적이다. 특히 본 연구에서 사용되고 있는 MonDB는 CAP(Consistency, Availability, Partition tolerance) 이론에 따라 데이터베이스를 분류하였을 때 Consistency와 Partition tolerance를 만족하는 CP형 데이터베이스로서, 현재 오픈소스 프로젝트로 진행되고 있으며, Key-Value의 방식으로 JSON형태의 문서데이터를 저장한다. 이는 스키마가 없으며, 정규표현 검색 및 배열 데이터의 특정값 포함여부 등의 검색조건 등에 유연하게 대응할 수 있다. 또한 NoSQL은 Scale out 방식의 수평적 확장(Horizontal Scalability)이 가능하여, 고가의 CPU, Memory 등의 업그레이드가 아닌, 기존 시스템과 동일한 장비를 병렬적으로 추가함으로써 저비용으로 시

스템을 확장할 수 있다. 따라서, 전통적인 RDBMS에 비하여 대량의 데이터를 병렬로 처리할 수 있으며 MapReduce 기법 등을 사용하여 데이터 클러스터링 연산, 통계, 데이터 추출 및 필터링이 가능하다.

감성(Sentiment)이란 “어떤 현상이나 일에 대하여 일어나는 마음이나 느끼는 기분이다[16]”. 감성 어휘는 객관적 가치 평가보다는 주로 내면이나 주관적 감정 또는 심리가 작용하는 의미영역을 묘사한 것이다. 감성 분석(Sentiment Analysis)은 자연언어처리와 전산언어학 그리고 텍스트 분석론을 활용하여 원자료에서 주관적인 정보를 발견하고 추출하는 과정이다[16]. 빅데이터로부터 사용자의 감성을 분석하기 위한 연구[17-19]가 진행되고 있다. 감성의 종류를 분석하고 분류하는 작업은 크게 세 가지의 단계로 나눌 수 있다. 첫 번째 단계는 감성 정보가 들어 있는 주관적인 생각이나 느낌을 표현하는 문장을 추출하고, 다음 단계에서 문서 또는 문장의 극성(긍정, 부정)을 나눈다. 마지막 단계는 문서 또는 문장이 어느 정도의 주관성을 갖는지 그 강도를 구하는 강도 분류이다[20-21]. 한국어의 감성분석 처리에는 영어 시소러스(Thesaurus) 및 영한사전을 이용한 극성 판별 방법 과 반자동/수동에 의해 구축된 의미사전을 이용한 방법이 활용된다. 이중 가장 정확도 및 활용도 측면에서 우수한 방법은 반자동/수동에 의한 의미사전 구축방법이 좋지만 비용이나 의미 분류 객관성의 측면에서의 문제점이 여전히 존재한다[22].

따라서, 본 연구에서는 다양한 비정형 SNS 데이터로부터 사용자의 감성정보를 추출할 수 있는 하둡(Hadoop) 시스템기반의 병렬적 HDFS(Hadoop Distributed File system)을 제안하고 HDFS시스템으로부터 전달받은 데이터로부터 사용자의 감성정보를 추출할 수 있는 맵리듀스(MapReduce)[23]기반의 감성분석함수를 제안한다.

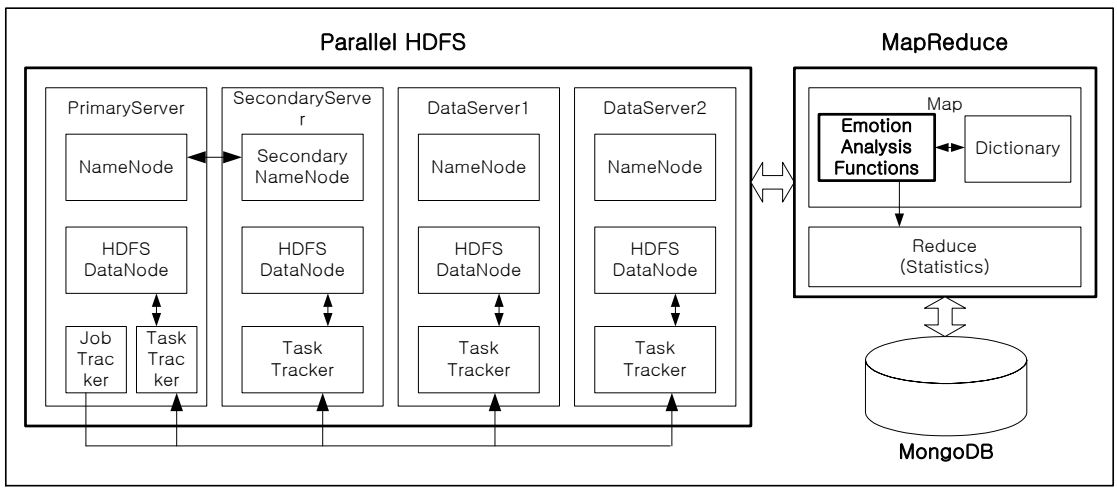
III. 비정형 감성정보 추출

3.1 병렬적 HDFS 구성

대량의 SNS 데이터로부터 비정형 데이터를 안정적으로 수집하여 text형태의 데이터를 추출하기 위한 병렬적 HDFS 시스템의 구성은 [그림 1]과 같다. HDFS시스템으로부터 추출된 text형태의 데이터는 감성정보 추출을 위한 다음 단계인 맵리듀스기반 감성분석함수의 입력으로 사용된다.

HDFS는 분산처리구조의 파일처리 시스템이다. [그림 1]과 같이 제안된 HDFS는 리눅스 기반의 4대의 서버로 병렬로 연결되며, 각각 데이터를 저장하기 위한 Node들의 chunk는 64MB로 구성되며, 장애 복구를 위해 NFS를 이용한 네임서버를 이중화한다. 구성된 서버의 기능은 [표 1]과 같다.

[그림 1] 제안시스템의 구성



[표 1] HDFS 서버의 기능

Server	Components	Role
PrimaryServer (Master Node)	Namenode, DataNode MapReduce, Crawler	Main server for parallel distribution process Name node(controlling other servers) Data node, Data loading
SecondaryServer (Slave Node 1)	Secondary NameNode DataNode	Backup server of main server Data node, Data loading
DataServer1 (Slave Node 2)	DataNode	Data node, Data loading
DataServer2 (Slave Node 3)	DataNode	Data node, Data loading

3.2 맵리듀스(MapRedude)기반 감성분석함수

맵리듀스는 분산 컴퓨팅을 지원하기 위한 목적으로 구글에서 개발한 소프트웨어 프레임워크로 맵(map)과 리듀스(reduce)라는 함수의 개념을 이용하여 병렬 프로그래밍을 가능하게 한다. 본 연구는 [표 2]와 같은 4가지 종류의 맵리듀스기반 감성분석함수를 제안한다. 즉, 긍정/부정 문맥 분석, 형태소 분석, 토근 분석, 금칙어 분석 함수이다.

[표 2] 제안된 감성분석 함수

emotion function	role	referenced dictionary
positive/negative context analysis function	context analysis using sentence pattern matching	positive/negative context dictionary
morphological analysis function	elimination of needless elements, calculation of the result count	positive/negative word dictionary
token analysis function	creation of tokens calculation of the result count	"
prohibited words analysis function	calculation of the prohibited word score	prohibited word dictionary

첫째, 긍정/부정 문맥 분석 함수이다. 이 함수는 먼저 정확도를 높이기 위해 한 문장 단위로 문맥을 검사하고, 긍정문맥 사전과 부정문맥 사전을 이용하여 패턴(정규식) 칭을 실시한 후, 원본 자료(트위트 text)를 긍정과 부정으로 카운팅하며, 긍정과 부정의 카운트가 동일하면 긍정으로 처리하고 판단 불가일 경우 형태소 분석으로 이관한다. 문맥분석을 위한 알고리즘은 [그림 2]과 같다.

둘째, 형태소 분석 함수이다. 이 함수에서는 한나눔의 한글형태소 분석기를 이용하여 링크, 특수기호 등 분석에 불필요한 요소를 제거한 후, 긍정어절과 부정어절 사전을 비교하여 각각의 카운터를 계산한다. 또한 긍정 또는 부정 카운터의 수치가 동일하다면 긍정으로 처리, 판단 불가한 상태라면 토큰분석으로 이관한다.

셋째, 토튼 분석 함수이다. 이 함수은 source(트위트 text)의 토큰을 공백으로 분리하고, 한글 형태소 분석기를 사용하여 형태소를 분석한 결과와 긍정어절과 부정어절 사전을 비교하여 각각의 카운터를 계산하며, 긍정 또는 부정카운터의 수치가 동일하다면 긍정으로 처리, 판단 불가한 상태라면 금칙어 분석으로 이관한다.

넷째, 금칙어 분석 함수이다. 이 함수에서는 최종 분석 단계로 상위의 과정에서 분석이 이루어지지 못했을 경우 금칙어 사전을 기반으로 금칙어스코어 계산 한다. 형태소 분석, 토근 분석 및 금칙어 분석을 위한 알고리즘은 [그

림 2]과 [그림 3]와 같다.

[그림 2] 문맥분석함수

```
//Context Analysis
1. input keyword, source
//keyword: target word for decision of positive or negative emotion
//source: source data of text form that processed by HDFS
  initialize result // a criteria for emotion decision
2. //pre-processing
   lower-case the keyword
   lower-case the source
   elimination of needless characters in source text
3.  initialize positive_count and negative_count
   //Context Analysis
   get the minimum sentence unit from the source
4.  //Computation of the positive_count and negative_count
   if minimum sentence unit = positive then positive_count++
   if minimum sentence unit = negative then negative_count++
   repetition step 4 until there is no minimum sentence unit
5. //Computation of the result by positive_count and negative_count
   if positive_count=0 and negative_count=0 then result=0(undecidable)
   if positive_count=negative_count then result=1(positive)
   result=positive_count-negative_count
```

[그림 3] 형태소분석함수

```
//Morphological Analysis - if result=0 in previous stage
1.
 1-1. input source
      initialize result-s //a criteria for emotion decision
 1-2. //pre-processing source
      elimination of needless characters in source text

 1-3. initialize positive_count_s and negative_count_s
 1-4. //Computation of the positive_count_s and negative_count_s using // the positive/negative word dictionary
      compute positive_count_s
      compute negative_count_s
    repetition step 1-4 until there is no morpheme unit
 1-5. if positive_count_s=0 and negative_count_s=0 then result-s=0
      if positive_count_s=negative_count_s then result-s=positive_count_s
      result-s=positive_count_s-negative_count_s
```

[그림 4] 토큰 및 금칙어 분석을 위한 알고리즘

```
//Token Analysis - if result-s=0 in previous stage
2.
 2-1. creation of tokens
 2-2. initialize positive_count_s and negative_count_s
 2-3. //Computation of the positive_count_s and negative_count_s using // the positive/negative word dictionary
      compute positive_count_s
      compute negative_count_s
    repetition step 2-3 until there is no token
 2-4 if positive_count_s=0 and negative_count_s=0 then result-s=0
      if positive_count_s=negative_count_s then result-s=positive_count_s
      result-s=positive_count_s-negative_count_s

//Prohibited word Analysis - if result-s=0 in previous stage
3.
 3-1. //Computation of the positive_count_s and negative_count_s using // the prohibited word dictionary
      compute positive_count_s
      compute negative_count_s
      result-s=positive_count_s-negative_count_s
```

맵리듀스 함수는 목적에 따라 [표 3]과 같은 모두 5가지 종류의 사전을 이용한다. 즉, 긍정어(em_positive), 부정어(em_negative), 긍정문맥(p_context_pattern), 부정문맥(n_context_pattern), 금칙어(abuses) 사전이다.

[표 3] 감성분석 사전의 역할

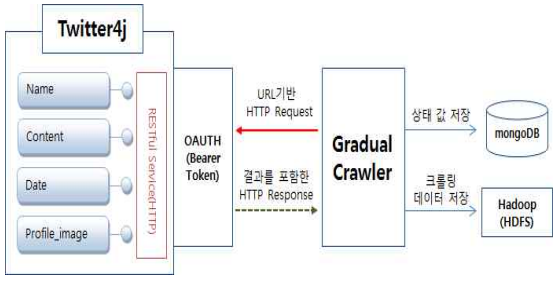
Dictionary	Role	application
Positive Context Dictionary	set of positive context patterns, compute the number of positive context in source sentence	Context Analysis
Negative Context Dictionary	set of negative context patterns, compute the number of negative context in source sentence	〃
Positive Word Dictionary	set of positive word patterns, compute the number of positive word in source sentence	Morphological/Token Analysis
Negative Word Dictionary	set of negative word patterns, compute the number of negative word in source sentence	〃
Prohibited Word Dictionary	set of prohibited words	Prohibited Word Analysis

IV. 실험 및 결과

4.1 비정형 SNS 데이터 수집

제안한 시스템의 성능분석을 위한 데이터 수집은 트위트(Twitter)를 통해 이루어졌다. 과거 데이터의 수집이 완료된 후 지속적인 증분 데이타의 경우 Twitter4j를 이용하여 트위터에서 제공하는 데이터를 수집한다. 트위터 제공 데이터는 현시점으로부터 최대 1주일 과거로의 데이터만을 수집 가능하며, 트위트에서 제공되는 키(토큰)는 15분 동안 450개의 쿼리를 사용할 수 있다. 본 연구에서는 크롤러(crawler)를 통해 매 4시간마다 데이터 수집 모듈을 실행하도록 하였다. [그림 5]는 Twitter4j를 이용하여 트위터로부터 데이터를 수집하였다.

[그림 5] Twitter4j를 통한 트위트 데이터 수집



4.2 실험 환경

제안 시스템의 성능분석을 위한 환경은 [표 4]와 같다. 실험환경은 4대의 서버를 하둡기반의 병렬시스템으로 구성하였으며, 사용 운영체제는 CentOS 6.3 x64를 사용하였다.

[표 4] 실험 환경

Components	Roles
OS, RE	Use of Hadoop for distributed storade, Supporting Java environment for processing some business logic
Crawler, HDFS Layer	Crawler: Gathering the source data from various SNSs HDFS: Distribution File system,
MapReduce Layer	Sentence Analysis, Text Mining, Emotion Analysis
MongoDB	Storing analyzed results by MapReduce in MongoDB
WAS, Web Server	Supporting Web applications using analyzed results

4.3 실험 분석 및 평가

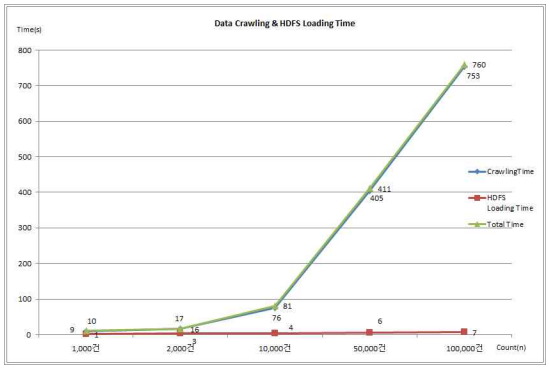
제안된 시스템의 성능분석을 위해 아래와 같은 실험을 진행하였다. 실험은 [표 5]와 같이 크롤러를 통해 수집된 5개 셋트의 실제 Twitter데이터에 대해 이루어졌다.

[표 5] 실험 데이터 추출

no.	Number of data	extraction period (day)	API
1	1,000	1	Twitter API
2	20,000	1	"
3	10,000	1	"
4	50,000	1	"
5	100,000	1	"

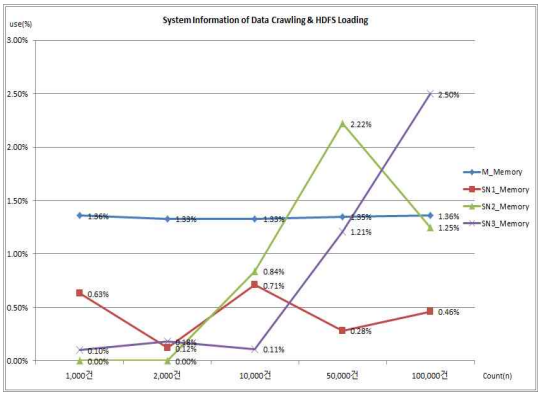
첫째, 크롤링(데이터수집) 및 HDFS적재에 관한 시스템 성능실험이다. [그림 6]은 데이터 셋트별 크롤링 시간 및 HDFS 적재시간을 비교한 것이다. [그림 6]과 같이 1,000건 데이터 셋트의 경우 크롤링 시간은 9초, HDFS 적재시간은 1초인 것으로 나타났고, 100,000건 데이터 셋트의 경우 크롤링 시간은 753초, HDFS 적재시간은 7초인 것으로 나타났다. 자료의 건수가 증가함에 따라 크롤링 시간과 HDFS적재 시간모두 증가하지만, HDFS시간에 비해 크롤링 시간이 절대적으로 많이 소요되는 것으로 나타났다. 따라서 제안된 HDFS에서 데이터량에 따른 데이터적재 부하는 미비함을 알 수 있다. 또한 제안된 시스템에서 크롤링과 HDFS적재는 시간복잡도 $O(n)$보다 빠르게 처리됨을 알 수 있다.

[그림 6] 크롤링 시간 및 HDFS 적재 시간



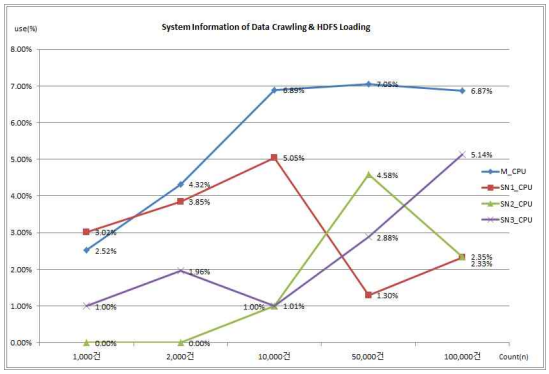
[그림 7]은 데이터 셋트별 크롤링 및 HDFS 적재시 각 노드별 메모리 부하를 나타낸 것이다. [그림 7]와 같이 슬레이브 노드 1(SN1)에서 슬레이브 노드 3(SN3)까지의 노드들은 메모리 사용량이 최소 0.10%에서 최대 2.50% 사용한 것으로 나타났고, 마스터 노드(M)의 경우 최소 1.33%에서 최대 1.36%를 사용한 것으로 나타났다. 데이터 분산 정책을 사용하는 마스터 노드의 경우 데이터 셋트의 규모에 상관없이 안정적인 메모리 사용을 나타내었고, Secondery 서버로 사용되는 슬레이브 노드 1도 비교적 안정적인 메모리 사용을 나타내었다. 다만 데이터 전용서버인 슬레이브 노드 2와 3은 데이터 셋트의 규모가 증가함에 따라 메모리 사용이 증가하였다. 또한 마스터 노드의 데이터 분산 적재 정책에 따라 특정 노드에만 부하가 가중되는 것이 아니라 부하가 균형적으로 분산되어 메모리가 효율적으로 사용됨을 알 수 있다.

[그림 7] 크롤링 및 HDFS 적재 시 노드별 메모리부하



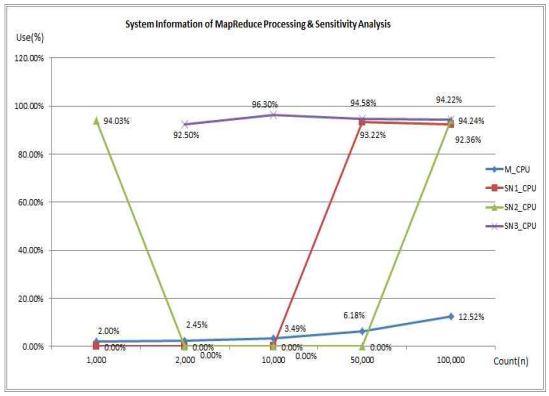
[그림 8]은 데이터 셋트별 크롤링 및 HDFS 적재시 각 노드별 CPU부하를 나타낸 것이다. [그림 8]과 같이 슬레이브 노드 1(SN1)과 슬레이브 노드 2(SN2)의 경우 최소 0.0%에서 최대 5.14%의 CPU 사용량을 나타냈다. 마스터 노드의 경우 최소 2.52%에서 최대 7.05%의 CPU 사용량을 나타내었다. 메모리 사용과 마찬가지로 CPU사용에 있어서도 특정 노드에 부하가 집중되는 것이 아니라 HDFS의 자동병렬처리 과정에서 슬레이브 노드간의 균형(balancing)이 이루어지는 것으로 나타났다. 마스터 노드의 경우도 데이터의 수집과 적재 시 안정적인 CPU 사용 환경을 제공하는 것으로 나타났다.

[그림 8] 크롤링 및 HDFS 적재 시 각 노드별 CPU부하



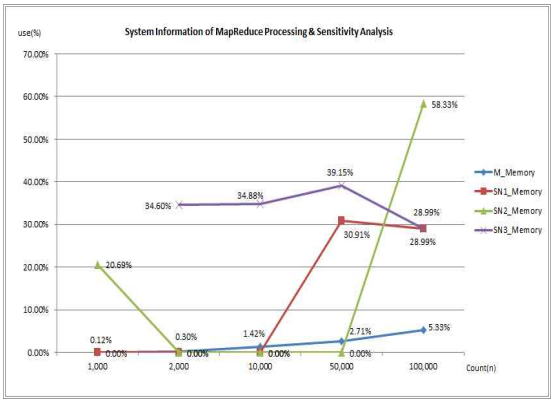
둘째, 맵리듀스 및 감성분석에 관한 시스템 성능실험이다. 첫 번째 실험에서 사용한 데이터 셋트를 가지고 맵리듀스 및 감성분석 시 시스템 부하정도를 실험하였다. [그림 9]과 [그림 10]는 맵리듀스 및 감성분석 시 각 노드별 CPU부하와 메모리부하를 각각 비교한 것이다. [그림 9]과 같이 마스트 노드(M)의 경우 실제 분석처리를 담당하지 않고 하위 슬레이브 노드들의 관리를 담당함으로써 CPU의 사용량이 낮은 반면, 슬레이브 노드들이 대부분의 CPU자원을 사용하는 것으로 나타났다. 데이터 건수 10,000인 데이터 셋트까지는 각 슬레이브 노드가 데이터를 상호병렬처리하는 것으로 나타났으나, 데이터 건수 50,000인 데이터 셋트 이후는 분산된 데이터량이 많아짐에 따라 모든 슬레이브 노드들이 CPU를 최대로 이용하는 것으로 나타났다. 따라서 제안된 시스템은 일정한 수준까지 안정적인 상호병렬처리가 이루어지는 것으로 볼 수 있다.

[그림 9] 맵리듀스 및 감성 분석 시 노드별 CPU부하



[그림 10]도 마찬가지로 마스트 노드의 경우 메모리 사용량은 낮으나 슬레이브 노드의 경우 상호병렬처리로 메모리가 분배되어 이용되어짐을 알 수 있다. 따라서 메모리 이용 측면에서도 슬레이브 노드 간에 적절한 상호병렬처리가 이루어지고 있음을 알 수 있다.

[그림 10] 맵리듀스 및 감성 분석 시 노드별 메모리부하



따라서, 제안된 시스템과 알고리즘을 이용하여 데이터를 수집하고 처리하였을 때 데이터 건수에 따라 적절한 처리시간을 나타냈으며, 자원할당 측면에서 단일 노드에만 시스템부하가 집중되지 않고 상호 병렬처리됨으로써 안정적인 병렬 분석 환경을 제공하는 것으로 나타났다.

V. 결론

본 연구에서는 SNS로부터 발생하는 대량의 비정형 데이터로부터 사용자의 감성을 분석할 수 있는 HDFS와 맵리듀스 함수를 제안하였다. 제안된 시스템은 하둡 시스템을 기반으로 병렬적으로 작동되는 4개의 주요서버로 구성되어 있으며, 4개의 주요기능을 가진 맵리듀스 함수로 구성된다. 제안된 시스템의 성능분석을 위해 몇 가지의 실험을 실시하였다. 즉, 데이터량에 따른 크롤링(데이터수집) 및 HDFS 적재에 관한 시스템 성능실험과 맵리듀스 및 감성분석에 관한 시스템 성능실험이었다. 실험을 통하여 제안된 HDFS는 데이터량의 변화에 따라 O(n)보다 적은 적재시간을 소모하였고, 시스템부하가 어느 하나의 노드에 집중되지 않고 각 노드에 병렬처리됨으로써 안정적인 성능을 보였다. 또한, 맵리듀스와 감성분석에 따른 성능실험에서 제안된 시스템과 맵리듀스 감성분석함수를 이용하여 데이터를 처리하였을 때 단일 노드에 부하가 집중되지 않고 상호 병렬처리 됨으로써 안정적인 병렬 분석환경을 제공한다.

참 고 문 헌

[1] McKinsey, 2011, “Big Data: The Next Frontier for Innovation, Competition, and Productivity”, [Online] McKinsey & Company, http://www.mckinsey.com/

[2] Chang-Shing Lee, Mei-Hui Wang, “Automated ontology construction for unstructured text documents”, Data & Knowledge Engineering, Vol.60, Iss.3, pp.547-566, 2007

[3] B. Lee, J. Lim, J. Yoo, “Utilization of Social Media Analysis using Big Data”, Jour. of the Korea Contents Association, Vol.13, No.2, pp.211-219, 2013

[4] M. Song, S. Kim, “A Study of improving on prediction model by analyzing method Big data”, The Journal of Digital Policy & Management, Vol.11, No.6, pp.103-112, 2013

[5] Ah Tan, “Text mining: The state of the art and the challenges”, Proc. of the PAKDD 1999, 1999

[6] Q. Mei, C. Xhai, “Discovering evolutionary theme patterns from text: an exploration of temporal text mining”, Proc. of the 11th ACM SIGKDD international conference on knowledge discovery in data mining, pp.198-207, 2005

[7] K. Park, K. Hwang, “A Bio-Text Mining System Based on Natural Language Processing”, Jour. of KISS: computing practices, Vol.17, No.4, pp.205-213, 2011

[8] B. Pang, L. Lee, “Opinion Mining and Sentiment Analysis”, Foundations and Trends in Information Retrieval, Vol.2, No.1-2, pp.1-135,

2008

[9] B. Kang, M. Song, “A Study on Opinion Mining of Newspaper Texts based on Topic Modeling”, Jour. of the Korean Library and Information Science Society, Vol.47, No.4, pp.315-334, 2013

[10] http://hadoop.apache.org/

[11] Jing Han, Kian Du, “Survey on NoSQL database”, Proc. of 6th International Conference on Pervasive Computing and Applications(ICPCA), pp.363-366, 2011

[12] Fay Chang, R.E. Gruber, “Bigtable: A Distributed Storage System for Structured Data”, ACM Transations on Computer System, Vol.26, Iss.2, 2008

[13] S. Sivasubramanian, “Amazon dynamoDB: a seamlessly scalable non-relational database service”, Proc. of the 2012 ACM SIGMOD'12, pp.729-730, 2012

[14] Lars George, “HBase: The Definitive Guide”, O'REILLY, 2011

[15] Kristina Chodorow, “MongoDB: The Definitive Guide 2nd Edition”, O'REILLY, 2013

[16] B. Pang, ,L. Lee, "Opinion Mining and Sentiment Analysis," Foundations and Trends in Information Retrieval: Vol.2, No.1-2,pp.1-135, 2008.

[17] S. Mukherjee, P. Bhattacharyya, “Sentiment Analysis in Twitter with Lightweight Discourse Analysis”, Proc. of COLING 2012, pp.1847-1864, 2012

[18] N. Godbole, S. Skiena, “Large-Scale Sentiment Analysis for News and Blogs”, Proc. of the ICWSM'2007, 2007

[19] A. Pak, P. Paroubek, “Twitter as a Corpus for Sentiment Analysis and Opinion Mining”, Proc. of the LREC'2010, 2010

[20] H. Tang, S. Tan, X. Cheng, " A survey on sentiment detection of reviews," Expert Systems with Applications, Vol.36, pp.10760-10773, 2009.

[21] Seth Gilbert, Nancy Lynch, Brewer's conjecture and the feasibility of consistent, available, partition-tolerant web services, ACM SIGACT New 33(2), pp. 51-59, 2002.

[22] E. Yu, Y. Kim, N. Kim, S. Jeong, “Predicting the Direction of the Stock Index by Using a Domain-Specific Sentiment Dictionary”, Jour. of Intelligence and Information Systems, Vol.19, No.1, pp.95-110, 2013

[23] J. Dean, S. Ghemawat, “MapReduce; Simplified Data Processing on Large Clusters”, Communications of the ACM, Vol.51, No.1, pp.107-113, 2008