

하둡 모델의 분석 및 보완 연구

이진우^o, 김수경^{*}

^o*한밭대학교 컴퓨터공학과

e-mail: ljw0084@nate.com^o, kimsk@hanbat.ac.kr^{*}

Complementary research and Analysis for hadoop

Jin-Woo Lee^o, Su-Kyoung Kim^{*}

^o*Dept. of Computer Engineering, Korea-HanBat National University

● 요약 ●

소셜 네트워크와 웹 2.0의 등장은 거대한 데이터 홍수를 초래하였다. 이 와 관련된 다양한 기술들이 연구 개발되고 있으며 특히 동시에 요구되는 data를 처리하기위한 여러 기술이 등장하였다. 본 연구에서는 다양한 BigData 분산처리 기술들중에 가장 각광 받고 있는 Hadoop이라는 기술을 연구 분석할 것이다. 국내에 아직 많은 사용자가 없어 그 존재감이 많이 없다가 요즘 들어 상승하고 있는 추세이며 이러한 Hadoop의 흐름속에 data의 분산과 병렬처리에서 발생하는 문제점을 분석하고 이를 해결할 수 있는 모델을 제시하여 새로운 모델의 하둡으로 기본적인 핵심기술인 federation을 쉽게 할 수 있고 향후 이 구조의 기능과 상세모델을 연구하고 구현하여 제안된 연구 구조의 우수성을 입증하고자 한다.

키워드: 하둡(Hadoop), 빅데이터(Big Data), 분산처리, federation

I. 서론

소셜 네트워크와 웹 2.0의 발전은 BigData의 처리에 대한 필요성을 증대 하였고 이에 따라 BigData와 관련된 다양한 기술들이 연구 개발되고 있다. 특히 동시에 요구되는 data를 처리하기 위해 최근에는 Hadoop을 기반으로 병렬분산 처리를 도입하고 있으나. 국내에서는 아직 Hadoop과 같은 플랫폼을 다뤄본 엔지니어들이 많이 있지 않다. 또한 아직 Hadoop에 대한 확실한 개념과 의식이 정립 되어 있지 않아 아직은 적용이 어려운 상황이다. 따라서 본 연구는 하둡의 모델에 있어 data의 분산과 병렬처리에서 발생하는 문제점을 분석하고 이를 해결할 수 있는 모델을 제시하여 Hadoop의 사용성과 성능을 높이고자 한다.

데 앞장섰다.

표 1. BigData 분석기법

분석기법	내용
Text Mining	자연어 처리에 기반을 두어 유용한 정보를 추출 가능. 의미있는 정보 추출 및 정보의 연계성 파악
Opining Mining	정형/비정형 텍스트의 긍정, 부정, 중립의 선호도를 판별. 서비스 및 상품에 대한 시장규모 예측, 소비자 반응분석에 사용
Social Network Analytic	그래프 이론에 뿌리를 두고 소셜 네트워크 연결구조 및 연결강도 등을 바탕으로 네트워크상의 허브(Hub) 역할을하는 사용자를 찾는데 사용
Cluster Analysis	비슷한 특성을 가진 개체를 합쳐가면서 최종적으로 유사 특성 군(group)을 발굴. 관심사에 따른 사용자군을 군집분석을 통해 분류

II. 관련 연구

1. Big Data 처리 기술

최근 소셜미디어 등 비정형 데이터의 증가로 인해, 분석 기법들 중에서 텍스트/오픈이언 마이닝, 소셜네트워크 분석, 군집분석 등이 주목을 받고 있다. 다음의 분석기법(표1)은 테라바이트 혹은 페타바이트 규모의 데이터에 적용되고 있다. 빅 데이터 분석을 수행할 수 있는 인프라 기술(표2)들로서 일찍이 트래픽을 점유해왔던 회사들은 빅데이터 처리를 위한 인프라 기술에 힘써왔으며 야후 아마존 구글 등의 회사들은 각자의 기술을 개발, 오픈소스화 하는

표 2. BigData 분석 처리 기술

분석기술	내용
Hadoop	분산처리기술 프로젝트로, 현재 빅데이터 분석에 가장 선호되는 솔루션이라고 할 수 있다. 주요 구성요소로 하둡 분산 파일 시스템인 HDFS, Hbase, MapReduce가 포함된다.
R	통계계산 및 시각화를 위한 언어 및 개발환경을 제공. R언어를 통해 통계기법부터 모델링, 최신 데이터 마이닝 기법까지 구현/개선이 가능. 통계분석 분야에서 높은 인지도를 가짐

NoSQL	비관계형 데이터베이스를 의미한다. 대표적인 NoSQL 솔루션으로는 Cassandra, Hbase, MongoDB 등이 존재한다. NoSQL은 스키마가 고정되지 않고 테이블간의 조인연산을 지원하지 않으며 수평적 확장이 용이하다. 분산가능성에 중점을 두고 있으나 하지만 안정성이 떨어진다.
-------	-----------------------------------------------------------------------------------------------------------------------------------------------------------------

2. Hadoop 의 처리 방법 분석

2.1 하둡의 구조

하둡 분산 파일 시스템(Hadoop Distributed File System, HDFS)는 기성 하드웨어에서 실행할 수 있도록 디자인된 분산 파일 시스템이다. HDFS는 상당히 고장에 강하고(fault tolerant) 저비용 하드웨어를 통해 배포가 가능하다.

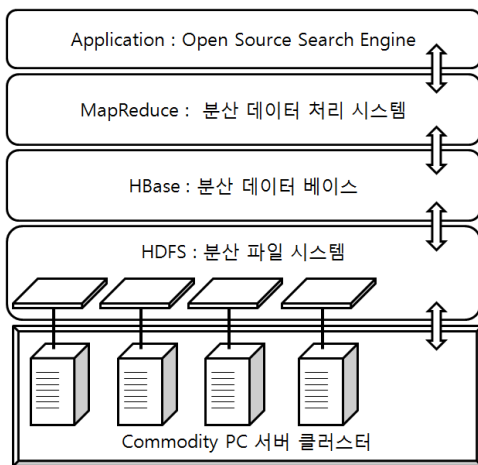


그림 1. 하둡의 구조(Architecture of Hadoop)

위의 (그림 1) 은 하둡의 전체적인 구조를 나타낸 것이다.

2.2 하둡 파일 분산 시스템 HDFS(Hadoop Distributed File System)

HDFS는 마스터/슬레이브 구조로 이루어져있다. 마스터 노드인 네임노드는 파일의 메타정보를 관리하고 클라이언트와 데이터노드 사이에서 I/O를 제어 하고 데이터 노드와 블록들의 맵핑을 결정한다(실제 파일시스템의 네임스페이스의 여러 기능을 수행). 슬레이브 노드인 데이터 노드는 단일 마스터 노드에 여러개가 달려 있고 하나의 데이터 노드는 일정크기(default 64MB)의 블록 단위로 나뉘어 관리되며 이 블록들을 여러개의 데이터 노드에 분산 및 복제 저장한다. 또 데이터 노드는 클라이언트가 요구하는 읽기 쓰기 기능을 담당한다.

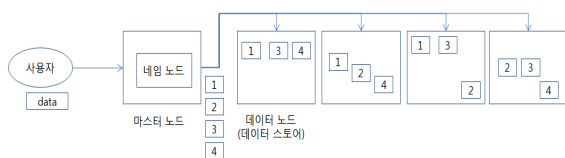


그림 2. HDFS의 구조

2.3 HBase

다음으로 HBase는 분산 데이터베이스이다. 이것은 기존에 RDBMS와 다르게 BigData를 분산처리 관리 하는데에 좀더 특화되어있는 데이터베이스이다. 기존 RDBMS가 2차원(row*col)의 공간을 제공했다면 HBase는 3차원(row*col*timestamp)의 공간을 제공한다. 또한 마스터 서버가 감당할 수 있을 만큼 자유롭게 확장할 수 있다. 낮은 값인 TimeStamp는 같은 데이터에 대한 여러 버전을 가질 수 있다.(예를 들어 최근 7일간의 데이터 값만 유지할 수 있다.)

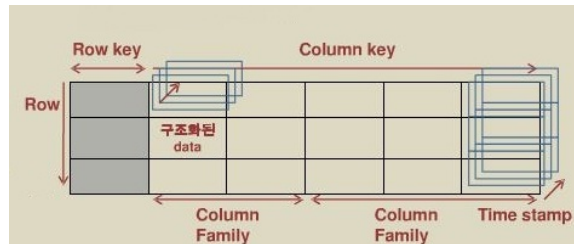


그림 3. HBase의 기본 모델

2.4 맵리듀스

마지막으로 맵리듀스는 맵 단계와 리듀스 단계로 처리과정을 나누어 작업한다. 맵과 리듀스 함수는 프로그래머가 직접 작성한다. 맵 함수는 분석대상의 정보들을 추출한다. 그리고 리듀스 함수로 가기전의 중간단계인 셔플(정렬)을 한다. 이후 사용자가 작성한 리듀스 함수에 맞춰 맵으로부터 온 정보를 입력받아 처리한다.

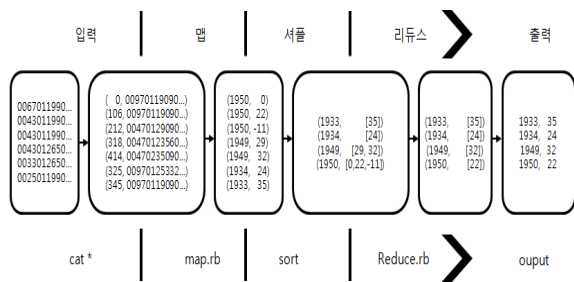


그림 4. 맵리듀스를 사용해서 연도별 최고기온 구하는 과정

III. 연구내용

1. 제안 내용 및 방법

기존의 RDBMS는 로컬에서의 탐색으로 서로다른 장소에 있는 컴퓨터 서버를 공유하여 하나의 데이터 저장소를 만들어 제공함으로써 하나의 저장소를 사용하는 것같은 정보를 주는 것이었고 하둡은 <그림 1>에서 보여주듯이 정보 탐색이 실제 물리적 저장소에 접근 하는 단계가 많고 각 단계간 data의 분산과 통합을 위한 네트워크가 필요하며 이를 위한 통신 방법이 중요하다.

기존 하둡 시스템 구조를 보면 하나의 마스터 노드(Job Tracker)

에서 여러 개의 슬레이브 노드(Task Tracker)로 Task를 분배하여 데이터를 병렬 처리한다. 이때에 슬레이브 노드들은 단일 마스터 노드를 처리하는 것이 아니라 처리가 끝남에 따라 같은 마스터 노드의 정보를 합쳐줘야 할 문제와 Task들의 종료 시간이 각기 다르다는 점에서 데이터 처리 안정성 문제가 제기 된다. (그림 5)는 위의 구조에 따라 NameNode와 DataNode들 간의 control과 Data Read/Write흐름을 도식화 한 것이다.

Task Set	DataNode (index)	NameNode (index)	flag	Time Stamp
Task 1	3	N1	Processing	12.04.05
Task 2	2	N1	Done	12.04.05
Task 3	8	N3	Done	12.05.10
Task 4	1	N4	Processing	12.05.12
:	:	:	:	
Task N	n	Nn	state	XX.XX.XX

그림 7. TaskSet의 구조

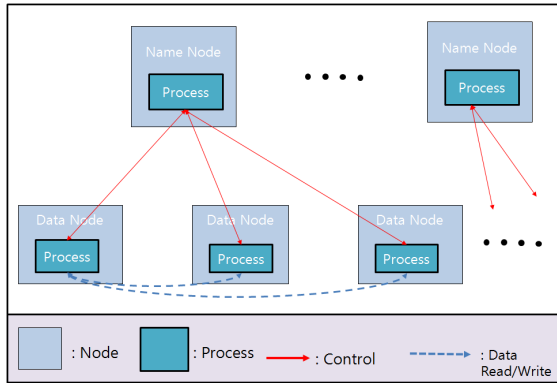


그림 5. 기존 하둡 시스템 구조

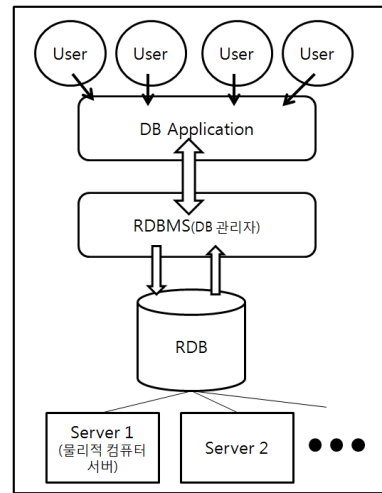


그림 8. 기존 DBMS 구조

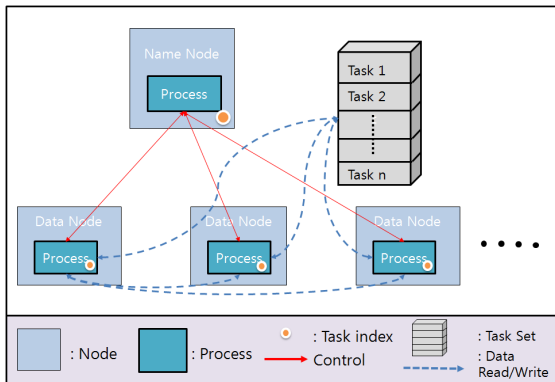


그림 6. 제안 개념 하둡 시스템 구조

위의 (그림 6)은 기존 하둡 시스템 구조를 보완한 제안된 개념이다. 기존에는 처리할 Task를 분할했다가 다시 합칠 때 그 Task의 출처를 찾는 어려움이 있었지만 본 연구는 처리할 Task에 색인(index)을 붙여서 DataNode에 분할되므로 Task가 할당되었더라도 이를 index를 기준으로 통합하기 때문에 합치는데 큰 어려움이 없다. 또한 각 Node별로 처리량이 다르고 처리 속도가 다르기 때문에 다른 Node들이 종료될때까지 기다리는 비효율적인 면이 있었지만 TaskSet을 두어 종료 됨과 동시에 처리된 정보를 색인에 맞게 담아두어 다른 Task를 처리할 수 있는 대기 상태로 만들어 DataNode를 더욱 효율적이게 활용 할 수 있게 했다. (그림 8)은 (그림 6)의 TaskSet의 구조를 설계한 것이다.

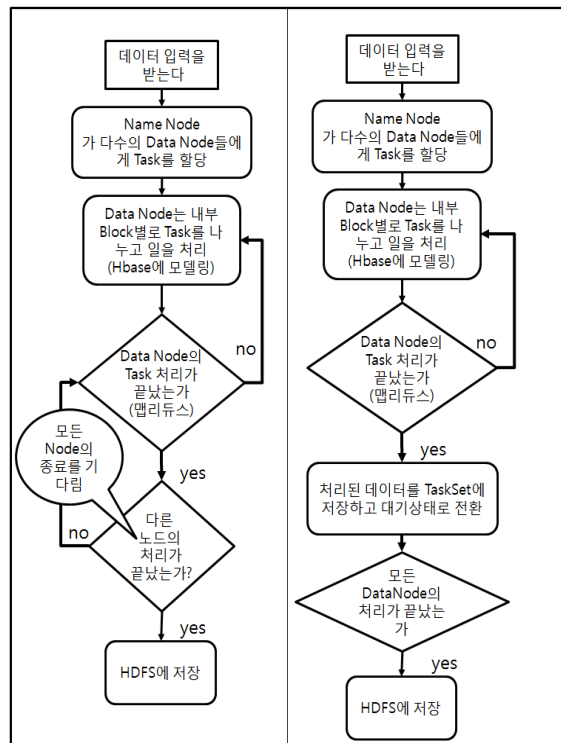


그림 9. 전체 처리 순서도(좌 기존, 우 제안개념)

IV. 결론

본 연구는 BigData의 분산처리 구조를 지원하는 Hadoop이 갖고 있는 문제점을 보완하고자 Taskset추가형 Hadoop구조를 제안하였다. 이 제안 구조에 따른 시스템의 구현과 실험은 진행되지 않았으나 빅 데이터의 연구의 필요성과 기술 개발의 중요성을 볼때 그 가치가 높다고 보인다. 향후 이 제안구조의 기능과 상세모형을 연구하고 구현하여 제안된 연구 구조의 우수성을 입증하고자 한다.

사사

이 논문은 2012년도 정보(교육과학기술부)의 재원으로 한국연구재단의 지원을 받아 수행된 기초연구사업임(No.20120004360)

참고문헌

- [1] <http://www.bloter.net/archives/104925>
- [2] <http://www.ciokorea.com/news/9928>
- [3] http://ko.wikipedia.org/wiki/%ED%95%98%EB%91%A1#cite_note-2
- [4] <http://agbird.egloos.com/5423801>, Hadoop_Guide_ext.pdf
- [5] Hadoop :The Definitive Guide(Tom White)
- [6] Parallel Algorithms Guy E. Belelloch and Bruce M.Maggs