The Journal of The Institute of Internet, Broadcasting and Communication (IIBC)
Vol. 15, No. 6, pp.17-24, Dec. 31, 2015. pISSN 2289-0238, eISSN 2289-0246

http://dx.doi.org/10.7236/JIIBC.2015.15.6.17
JIIBC 2015-6-3

빅 데이터를 이용한 스마트 응용의 설계

Design of a Smart Application using Big Data

오선진*

Sun-Jin Oh*

요 약 정보 기술과 첨단 무선 네트워크 응용 기술의 급속한 발전과 더불어, 방대하고 다양한 형태의 데이터들이 시시각각 양산되고 있으며, 최근 빅 데이터 분석기술의 중요성과 가치는 점차 증대되고 있다. 과거에는 너무 방대하여 관리조차 힘들어 무용지물이던 빅 데이터는 데이터 수집 컴퓨팅 장비와 분석 도구의 발전을 통해 다양한 활용분야에서 작은 규모의 데이터로는 불가능했던 새로운 영감이나 가치를 추출해 내는 것이 가능하게 되었다. 하지만 현실 세계에서는 아직도 빅 데이터 대부분이 제대로 적절하게 분석되어 사용되지 못하고 사장되는 것이 사실이다. 결국, 빅 데이터에서 통찰력 습득과 새로운 가치 창출을 위한 전제 조건으로 효율적인 빅 데이터 처리를 위한 분석 기술의 확보가 중요하다고 할 수 있다. 본 논문에서는 이러한 빅 데이터를 보다 효율적으로 처리하고 원하는 관심 정보를 효과적으로 추출해 낼 수 있는 정밀한 분석기법과 처리 기술을 연구하고 이를 실제 적용하는 스마트 응용을 설계한다.

Abstract With the rapid growth of Information technology and up-to-date wireless network application technologies, huge and various types of data are produced in every moment, the value and significance of the analysis techniques using big data are increased recently. Big data, which were useless since they were too huge to manage in the past, enables us to get new inspirations and values in various practical application areas through the development of big data computing devices and analytic tools. Nowadays, however, it is true that most of the big data are still wasted without properly analyzed and used. In the long run, the preliminary stipulations for finding inspirations and extracting new values from big data are securing big data analysis and application techniques to process big data efficiently. In this paper, we study accurate data analysis techniques and data process technologies those are able to extract needed inspirations and values from big data efficiently, then design the smart application that adopts these techniques practically.

Key Words : Data Analytic Technique, Big Data, Smart Application

Ⅰ. 서 론

지금 우리는 정보화 사회를 지나 '스마트 혁명'이라 불리는 스마트 시대에 살고 있다. 스마트 시대는 스마트 기술과 데이터의 창조적 활용을 통해 인간 중심의 스마트 가치를 실현하는 시대를 말한다. 스마트 시대를 이끌어 갈 핵심 키워드로는 단연 "빅 데이터"를 들 수 있으며 이는 큰 규모의 데이터를 활용해서 작은 규모의 데이터로는 불가능했던 새로운 통찰이나 다양한 형태의 가치를 추출해내는 활용 기술을 의미한다.[1] 최근 디지털 기술과 인터넷의 발달, 스마트 폰과 디지털 카메라 같은 대량 정보를 생산할 수 있는 모바일 기기의 보급, 온라인 상거래의 증가, 소셜 미디어와 SNS 등 소셜 네트워크 이용 확대 등으로 인해 생산되는 데이터 량이 기하급수적으로

*종신회원, 세명대학교 정보통신학부
접수일자: 2015년 10월 5일, 수정완료: 2015년 11월 5일
게재확정일자: 2015년 12월 11일

Received: 5 October, 2015 / Revised: 5 November, 2015 / Accepted: 11 December, 2015
*Corresponding Author: sjoh@semyung.ac.kr
Dept. of Computer & Information Science, Semyung University, Korea

증가하게 된 것이다. 빅 데이터란 이렇듯 데이터의 양, 생성주기, 형식 등에서 과거 데이터에 비해 규모가 크고, 형태가 다양하여 기존 방법으로는 수집, 저장, 검색, 분석이 어려운 방대한 크기의 데이터를 의미하며 일반적인 데이터베이스 체계가 저장, 관리 분석할 수 있는 범위를 초과하는 규모의 데이터를 말한다.[2] 빅 데이터는 과거에는 너무 방대하여 관리조차 힘들어 무용지물이었으나 데이터 수집, 병행 처리 장비 및 데이터 분석 도구의 발전을 통해 버려지던 데이터로부터 새로운 통찰과 가치있는 지식을 발견할 수 있게 되었다. 오늘날 빅 데이터의 중요성과 가치는 점점 증대되고 있으며 현재 활용분야도 과학기술 분야에서 점차적으로 공공 및 사업, 서비스 전 분야로 확산됨으로써 새로운 데이터 생태계를 조성해 나가고 있다. 빅 데이터는 정치, 사회, 경제, 문화, 과학기술 등 전 영역에 걸쳐서 사회와 인류에게 가치 있는 정보를 제공할 수 있는 가능성을 제시한다.[3] McKinsey는 빅 데이터의 사회 경제적 가치를 산업의 투명성 증대, 소비자 니즈 발견 • 트랜드 예측 • 성과 향상을 위한 실험, 소비자 맞춤형 비즈니스를 위한 고객 세분화, 자동 알고리즘을 통한 의사결정 지원과 대행, 비즈니스 모델 • 상품 • 서비스 혁신 등 다섯 가지로 제시하고 있다.[4]

하지만 현실 세계에서는 아직도 빅 데이터가 제대로 적절하게 응용되지 못하고 그냥 사장되어 버리고 있는 실정이다. 빅 데이터 분석이 기존의 대용량 데이터베이스 처리 방법과 다른 점은 대용량 데이터베이스는 사전에 미리 정의된 정형화된 데이터들의 집합으로 데이터가 분산되지 않고 한곳에 모인 데이터 집합을 대상으로 하였다. 따라서 대용량 데이터베이스는 주로 디스크 기반으로 구현되고 데이터는 주기적으로 갱신되며 이들의 분석은 패턴 매칭이나 빈도수, 평균 등의 통계분석이 주가 된다. 반면 빅 데이터는 정형 데이터뿐만 아니라 비정형화된 데이터를 포함한다. 데이터가 한곳에 집중되어 있기 보다는 여러 곳에 흩어져 있는 경우가 더 많으며, 스트리밍 방식의 데이터 갱신이 일어난다. 또한 효율적인 처리를 위해서는 디스크 기반보다는 in-memory 기반의 환경이 더 효과적이다. 빅 데이터 분석은 패턴이나 통계에 기반한 대용량 데이터베이스 처리와는 달리 인공지능, 기계학습, 의미 기반의 정보처리기법 등 다양한 분석 기법을 포함한다.[5] 성공적인 빅 데이터 활용을 위해서는 데이터의 자원화, 데이터를 가공하고 분석 • 처리하는 기술, 데이터의 의미를 통찰하는 인력 등 3가지 분야의 전략 수립이 필수적이다. 결국, 빅 데이터에서 통찰력 습득 및 새로운 가치를 창출하기 위한 필수 전제 조건으로서 빅 데이터 처리를 위한 기반 분석기술 개발과 스마트 응용기술 확보가 무엇보다도 중요하다고 할 수 있다.[6]

본 논문에서는 병렬처리 기반의 빅 데이터 처리 시스템을 구축하고 수집된 빅 데이터로부터 필요로 하는 새로운 통찰과 가치가 될 수 있는 유용한 정보를 효율적으로 결정하고 추출해 낼 수 있는 정밀한 데이터 분석방법을 연구하고, 이를 실제 적용할 수 있는 빅 데이터 기반 스마트 응용 모델을 제안하고 설계한다. 본 논문의 구성은 다음과 같다. 2장에서는 관련 연구를 살펴보았고, 3장에서는 빅 데이터를 이용한 스마트 응용을 위한 시스템 모델을 서술하였으며, 4장에서는 빅 데이터 기반 스마트 응용을 설계하였고, 5장에서는 설계한 스마트 응용에 대한 구현 결과와 고찰을 하였으며, 마지막으로 6장에서 향후 연구내용과 함께 결론을 맺는다.

II. 관련 연구

빅 데이터란 기존의 관리, 분석 체계로는 감당하기 어려운 막대한 량의 데이터 집합과 이를 해결하기 위한 플랫폼, 분석 기법 등을 포괄하고 있다. 따라서 빅 데이터 분석 기법들은 통계학, 전산학, 기계학습에서 응용되고 있으며 새로이 등장하는 빅 데이터 분석기술로는 통계기법, 데이터 마이닝, 기계학습, 자연어 처리, 패턴 인식, 소셜 네트워크 분석, 비디오 • 오디오 • 이미지 프로세싱 등이 이에 해당되며, 최근 빅 데이터와 관련하여 다음의 기술들이 새롭게 등장하였다.[7]

- 대용량 데이터 처리 능력을 위한 분산 처리 기술로 하둡(Hadoop), 분산 파일시스템(HDFS), 분산 데이터베이스 (HBase), MapReduce 등
- 인 메모리(In-Memory) 기술 : 인 메모리 기술에서는 메모리상에 필요한 데이터와 이의 인덱스를 보관함으로써 데이터 검색 시간을 크게 줄일 수 있음.
- 의미 분석 기술과 진보된 알고리즘 및 데이터 마이닝 기술: 예를 들어 통계 계산 및 그래픽을 위한 'R'언어 등.
- 비정형 데이터를 처리하기 위한 NoSQL 기술 : 아파

치 Cassandra와 CouchDB, 구글의 BigTable, 아마존 다이나모(Dynamo) DB, IBM Lotus Domino 등.

최근 빅 데이터 분석의 중요성이 부각되면서 과거의 데이터 마이닝 기법이 다시금 각광을 받고 있다. 데이터 마이닝 기법은 저장된 방대한 양의 데이터에서 자동으로 체계적이고 통계적인 규칙이나 패턴을 찾아내는 DB 기술로 KDD(Knowledge Discovery in Databases) 라고도 한다. 데이터 마이닝 기법은 통계학 분야에 기초를 둔 탐색적 데이터 분석, 가설 검증, 다변량 분석, 시계열 분석, 일반 선형모형 등의 방법론과 데이터베이스 분야에 기초를 둔 온라인 분석처리 (OLAP: Online Analytic Processing)기법, 인공지능 분야에 기초를 둔 자기조직화 지도(SOM: Self-Organizing Map), 신경망, 전문가 시스템 등 기술적 방법론 등을 사용한다. 데이터 마이닝 기법을 기반으로 하는 분석방법은 분류(classification), 예측(forecasting), 시계열 분석(time series analysis), 회귀분석(regression), 군집화(clustering), 연관규칙(association rule), 요약(summarization), 연속성(sequencing)과 같은 기술을 적용하여 의미 있는 결과를 도출해 낼 수 있다.[4, 8]

데이터 마이닝 기법에 기초한 분석 기술들이 다양하게 개발되어 최근 빅 데이터를 처리하고 분석하는 변형 기법이나 분석기법으로 활용되고 있다. 그 대표적인 응용 사례로는 Web Mining 기술[9], Text Mining 기술[10], Clustering 기술[11], Reality Mining 기술[12], Graph Mining 기술[13], Social Network Analysis 기술[14], Opinion Mining 기술[15] 등이 있으며 주로 빅 데이터에 포함된 사안이나 인물, 이슈, 이벤트 등에서 새로운 통찰이나 가치있는 의미를 추출하여 사람들의 의견이나 평가 등을 분석하는 기술로 많이 활용되고 있다.

Ⅲ. 시스템 모델

빅 데이터를 이용한 응용은 최근 여러 분야에서 다양하게 이용되고 있으나 이들 대부분은 방대한 양의 빅 데이터로부터 자주 출몰하는 키워드를 가려내서 그 빈도수를 이용하여 최근 경향이나 속성을 판단하는 응용에 그치고 있다. 이렇다 보니 실제 빅 데이터 처리를 위한 분석 도구로는 단순하게 빅 데이터에 포함되어 있는 특정

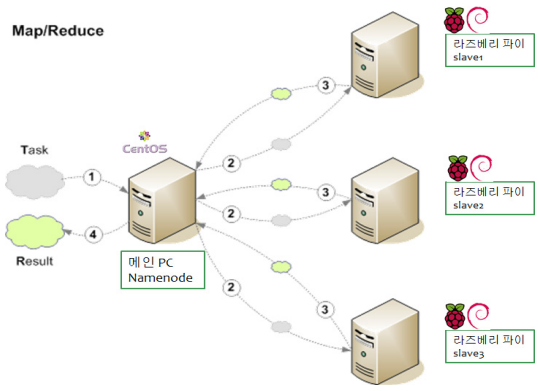

그림 1. 시스템 모델
Fig. 1. System Model

키워드를 인지하고 그 발생 빈도를 알기 위해 카운터를 이용한 간단한 통계처리 수준에 머물러 있다고 해도 과언이 아니다. 하지만 보다 정밀한 데이터 분석이나 의사결정을 필요로 하는 스마트 응용을 위해서는 이러한 수준의 통계 기법만으로는 좋은 결과를 기대하기 힘들다. 따라서 보다 정확하고 유연한 데이터 분석과 의사결정을 가능하게 하기 위해선 결정에 영향을 미칠 수 있는 결정요인을 복합적으로 분석할 수 있는 의사결정구조를 갖춘 데이터 분석 모델이 필요하다.

본 논문에서는 이러한 복합적인 의사결정모델을 기반으로 하는 스마트 응용을 제안하고 설계한다. 본 논문에서 고려하는 스마트 응용은 “빅 데이터를 이용한 항공기 연착 예측 시스템”으로 운항중인 여객기의 과거 비행관련 빅 데이터를 기반으로 항공기의 출발과 도착 지연을 야기했던 요인들을 분석하고 이를 바탕으로 현재 운항되는 항공기에 대한 연착을 예측하여 지연 가능성이 있는 비행 운항 스케줄을 인지하여 경고와 여행객의 피해를 최소화하는 비행 운항 스케줄로 조정을 수행하는 항공기 연착 예측 시스템을 설계하였다.

그림 1은 본 논문에서 설계한 빅 데이터를 이용한 항공기 연착 예측 시스템의 시스템 모델을 보여준다. 그림에서 보인바와 같이 대용량 데이터의 효율적인 처리를 위해 분산 메커니즘을 사용하여 데이터의 처리 속도를 향상시켰으며 하나의 메인 서버에 라즈베리 파이를 기반으로 데이터 마이닝을 수행하는 분산 서버를 망으로 연결하여 사용하였다. 메인 서버는 CentOS를 그리고 분산 서버인 라즈베리 파이(Raspberrypi)는 라즈비안을 기본 운영체제로 구축하였고 빅 데이터 처리를 위해 Hadoop-2.5.2 프로그램을 사용하였으며, 응용 프로그램

개발은 Java-7-openjdk를 기반으로 하는 Eclipse 통합 개발 환경을 사용하였고, 빅 데이터의 정확한 분석을 위해 Map-Reduce를 이용하여 처리하였다.

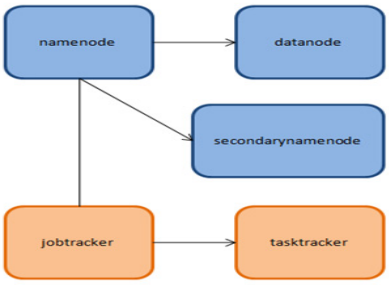

그림 2. 시스템 구성도
Fig. 2. Diagram of System Components.

그림 2는 빅 데이터를 이용한 항공기 연착 예측 시스템의 주요 시스템 구성도를 보여준다. 시스템을 구성하는 주요 컴포넌트들은 다음의 기능을 수행한다.

NameNode : HDFS의 모든 메타데이터를 관리하고, 클라이언트가 HDFS에 저장된 파일에 접근하게 한다.

HDFS (Hadoop Distributed File System) : 대용량 파일을 분산된 서버에 저장하고, 많은 클라이언트가 빠르게 처리할 수 있게 설계된 분산 파일 시스템이다.

DataNode : HDFS에 데이터를 입력하면 입력 데이터는 32MB의 블록으로 나눠져서 여러 개의 데이터노드에 분산되어 저장된다.

SecondaryNameNode : 보조 네임노드로 주기적으로 네임노드의 파일 시스템 이미지 파일을 갱신 수행한다.

Jobtracker : 하둡 클러스터에 등록된 전체 잡의 스케줄링을 관리하고 모니터링 한다.

TaskTracker : 맵-리듀스 프로그램을 실행하며, 하둡의 데이터노드에서 실행되는 데몬이다.

Ⅳ. 빅 데이터 스마트 응용 설계

이 장에서는 제안한 빅 데이터를 이용한 항공기 연착 예측 시스템을 설계하고 주요 알고리즘을 분석하였다. 최근 항공기를 이용한 국내외 여행객 수가 급속히 늘어나면서 공항에서의 항공기 트래픽이 급증하게 되었고 이로 인한 크고 작은 비행기 이착륙 지연이 빈번하게 발생

번호	칼럼 이름	내용
1	Year	연도, 1987 ~ 2008
2	Month	월, 1 ~ 12
3	DayofMonth	일, 1 ~ 31
4	DayOfWeek	요일, 1(월요일) ~ 7(일요일)
5	DepTime	실제 출발 시각, 현지 시각 기준 hhmm 형태로 표기
6	CRSDepTime	예정 출발 시각, 현지 시각 기준 hhmm 형태로 표기
7	ArrTime	실제 도착 시각, 현지 시각 기준 hhmm 형태로 표기
8	CRSArrTime	예정 도착 시각, 현지 시각 기준 hhmm 형태로 표기
9	UniqueCarrier	항공사 코드
10	FlightNum	항공편 번호
11	TailNum	항공기 등록 번호 (비행기 꼬리 날개 쪽에 표기)
12	ActualElapsedTime	실제 경과 시간, 분으로 표기
13	CRSElapsedTime	예정 경과 시간, 분으로 표기
14	AirTime	방송 시간, 분으로 표기
15	ArrDelay	도착 지연 시간, 분으로 표기
16	DepDelay	출발 지연 시간, 분으로 표기
17	Origin	출발지 공항 코드, IATA(국제 항공 운송 협회) 기준
18	Dest	도착지 공항 코드, IATA(국제 항공 운송 협회) 기준
19	Distance	비행 거리, 마일 기준
20	TaxiIn	비행기 바퀴가 지면에 닿아서(착륙) 목적지 공항의 게이트에 도착할 때까지 시간
21	TaxiOut	출발지 공항의 게이트에서 출발해서 바퀴가 지면에서 떨어질 때(이륙)까지의 시간
22	Cancelled	비행 취소 여부 -> 1: 예, 0: 아니오
23	CancellationCode	비행 취소 코드 -> A: 항공사, B: 기상, C: NAS(National Airspace System), D: 보안
24	Diverted	우회 여부 -> 1:예, 0: 아니오
25	CarrierDelay	항공사 지연 시간, 분으로 표기
26	WeatherDelay	기상 지연 시간, 분으로 표기
27	NASDelay	NAS 지연 시간, 분으로 표기
28	SecurityDelay	보안 지연 시간, 분으로 표기
29	LateAircraftDelay	연착 항공기 지연 시간, 분으로 표기

그림 3. 항공 운항 데이터 구조
Fig. 3. Structure of the Flight Operation Data

하고 있는 상황이다. 이러한 항공기의 연착은 승객의 불편을 초래할 뿐만 아니라 관제탑 등 공항 관계자나 탑승 승무원에 이르기까지 지연으로 인한 불편과 피해는 지속적으로 전파되어 그 여파가 눈덩이처럼 불어나게 된다. 본 시스템은 이러한 연착으로 인한 피해를 최소화 하도록 하기 위해 항공기 연착을 유발했던 과거의 비행 운항 자료를 기반으로 우리가 필요로 하는 새로운 통찰과 가치가 되는 유용한 정보를 효율적으로 결정하고 추출해 낼 수 있는 정밀한 복합 의사결정모델을 제안하고, 이를 적용하는 빅 데이터 기반 항공기 연착 예측 시스템을 설계한다. 그림 3은 본 논문에서 고려한 항공기 연착 예측 시스템에서 사용한 비행 운항 빅 데이터의 일부를 보여준다. 이 정보는 2005년부터 약 5년간의 비행 운항 정보 약 3000여만 건으로 다양한 필드의 방대한 양의 데이터를 포함하고 있다. 데이터를 구성하는 필드의 속성을 보면 크게 3개 부류로 나눌 수 있는데 첫 번째 부류는 비행 운항 내부 상황 정보인 항공기 기종, 편명, 수령, 이착륙 및 연착관련 정보, 비행 관련 기장을 포함한 승무원에 대한 정보를 포함한다. 두 번째 부류로는 항공기 이용 상황 정보로 탑승객 예약 및 탑승 상황정보, 비행 날짜, 요일, 시즌 정보(연휴, 성수기, 비수기 등)를 포함한다. 세 번째 부류는 비행 운항 외부상황 정보로 비행 당일의 일기와 공항 날씨정보, 비행 조건, 비행에 영향을 미칠 수 있는 특이 상황정보 등을 포함하게 된다. 항공기 연착을 유발시킨 요인 분석을 위해 방대한 비행 운항 정보로부터 주

요 관심사인 연착을 유발했던 정보만을 추출하여, 이를 세 부류로 구분한 비행 운항 내부 상황정보, 항공기 이용 상황정보 그리고 비행 운항 외부 상황정보에 대한 의사결정트리를 구성한다. 이들 결정트리로 부터 식(1)을 이용하여 각각 항공기 연착을 초래한 결정 요인에 대한 평균값을 구할 수 있다.

$$D_k = \frac{\sum_{i=1}^{m} d_i}{m} \tag{1}$$

여기서, D_k는 결정트리에서 각 등급별 항공기 지연을 초래한 결정요인의 평균을 나타내며, d_i는 결정 등급에 해당하는 각 항목의 결정값을 나타내며 m은 결정 등급에 해당되는 항목의 총 개수를 의미한다. 각각의 결정트리와 세 부류에서의 비행 운항 지연시간과의 상관관계를 알기 위해 이 평균값을 사용하여 이들 간의 회귀분석을 통하여 한 결정 요인이 다른 결정 요인에 어떻게 영향을 미치고 어느 정도의 영향이 예측되는지를 알아보게 된다. 본 논문에서는 비행기 연착을 유발하는 결정요인들 간의 상관도가 비교적 단순한 단순 회귀모형을 이용한 선형 회귀분석을 적용하여 첫 번째 부류에서는 항공기 기종 수령, 두 번째 부류에서는 운항 시즌정보 그리고 마지막 세 번째 부류에서는 비행 조건 정보를 수치화하여 입력변수로 하고 목표변수인 예측 지연시간과의 관계를 식(2)를 이용하여 추정하였다.

$$T_k = \alpha + \beta D_k + \epsilon \tag{2}$$

여기서 T_k는 각 부류에서의 연착 예측시간을 의미하며 α와 β는 회귀 계수이고 ϵ은 기대값 0과 분산 σ^2을 가지는 오차 항으로 단순 선형 회귀 모형의 모수 α와 β를 추정하기 위해서 least square method를 사용하여 오차들의 제곱합이 최소가 되도록 회귀 계수를 추정하는 방법을 사용하였다. 이들 세 부류의 정보가 항공기 연착에 미치는 영향은 그 정도에 따라 각각 상이하게 작용하므로 각 부류의 연착 예측시간들은 식(3)과 같이 연착 예측시간에 서로 다른 가중치를 부여한 조정 연착 예측시간을 사용하였다.

$$T = \frac{\sum_{i=1}^{3} \omega_i T_i}{3} \tag{3}$$

여기서 T는 조정 항공기 연착 예측시간, T_i는 각 부류에서의 연착 예측 시간 그리고 ω_i는 각 부류에서의 항공기 연착시간에 영향을 미치는 가중치를 의미한다.

```
INFO mapred.JobClient:  map 100% reduce 100%
INFO mapred.JobClient: Job complete: job_201507271811_0001
INFO mapred.JobClient: Counters: 29
INFO mapred.JobClient:   Map-Reduce Framework
INFO mapred.JobClient:     Spilled Records=54865138
INFO mapred.JobClient:     Map output materialized bytes=290126481
INFO mapred.JobClient:     Reduce input records=21900551
INFO mapred.JobClient:     Virtual memory (bytes) snapshot=202242461696
INFO mapred.JobClient:     Map input records=59285466
INFO mapred.JobClient:     SPLIT_RAW_BYTES=9768
INFO mapred.JobClient:     Map output bytes=246324051
INFO mapred.JobClient:     Reduce shuffle bytes=290126481
INFO mapred.JobClient:     Physical memory (bytes) snapshot=16719310848
INFO mapred.JobClient:     Reduce input groups=108
INFO mapred.JobClient:     Combine output records=0
INFO mapred.JobClient:     Reduce output records=108
INFO mapred.JobClient:     Map output records=21900551
INFO mapred.JobClient:     Combine input records=0
INFO mapred.JobClient:     CPU time spent (ms)=231690
INFO mapred.JobClient:     Total committed heap usage (bytes)=14150701056
INFO mapred.JobClient:   File Input Format Counters
INFO mapred.JobClient:     Bytes Read=5733396812
INFO mapred.JobClient:   FileSystemCounters
INFO mapred.JobClient:     HDFS_BYTES_READ=5733406580
INFO mapred.JobClient:     FILE_BYTES_WRITTEN=720219957
INFO mapred.JobClient:     FILE_BYTES_READ=436166045
INFO mapred.JobClient:     HDFS_BYTES_WRITTEN=1538
INFO mapred.JobClient:   Job Counters
INFO mapred.JobClient:     Launched map tasks=88
INFO mapred.JobClient:     Launched reduce tasks=1
INFO mapred.JobClient:     SLOTS_MILLIS_REDUCES=483382
INFO mapred.JobClient:     Total time spent by all reduces waiting after reserving slots (ms)=0
INFO mapred.JobClient:     SLOTS_MILLIS_MAPS=843486
INFO mapred.JobClient:     Total time spent by all maps waiting after reserving slots (ms)=0
INFO mapred.JobClient:     Data-local map tasks=88
INFO mapred.JobClient:   File Output Format Counters
INFO mapred.JobClient:     Bytes Written=1538
```

그림 4. 결정트리 구축을 위한 하둡 시스템의 빅데이터 처리

Fig. 4. Big Data Processing in Hadoop System for Constructing a Decision Tree.

```
1  package hadooparr;
2
3  import org.apache.hadoop.conf.Configuration;
4  import org.apache.hadoop.fs.Path;
5  import org.apache.hadoop.io.IntWritable;
6  import org.apache.hadoop.io.Text;
7  import org.apache.hadoop.mapreduce.Job;
8  import org.apache.hadoop.mapreduce.lib.input.FileInputFormat;
9  import org.apache.hadoop.mapreduce.lib.input.TextInputFormat;
10 import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;
11 import org.apache.hadoop.mapreduce.lib.output.TextOutputFormat;
12
13 public class ArrivalDelayCount {
14     public static void main(String[] args) throws Exception {
15         Configuration conf = new Configuration();
16
17         //입출력 데이터 경로 확인
18         if (args.length != 2) {
19             System.err.println("Usage: DepartureDelayCount <input> <output>");
20             System.exit(2);
21         }
22         // Job 이름 설정
23         Job job = new Job(conf, "DepartureDelayCount");
24
25         //입출력 데이터 경로 설정
26         FileInputFormat.addInputPath(job, new Path(args[0]));
27         FileOutputFormat.setOutputPath(job, new Path(args[1]));
28
29          // Job 클래스 설정
30          job.setJarByClass(ArrivalDelayCount.class);
31          // Mapper 클래스 설정
32          job.setMapperClass(ArrivalDelayCountMapper.class);
33          // Reducer 클래스 설정
34          job.setReducerClass(DelayCountReducer.class);
35          // 입출력 데이터 포맷 설정
36          job.setInputFormatClass(TextInputFormat.class);
37          job.setOutputFormatClass(TextOutputFormat.class);
38
39          //출력키 및 출력값 유형 설정
40          job.setOutputKeyClass(Text.class);
41          job.setOutputValueClass(IntWritable.class);
42
43          job.waitForCompletion(true);
44     }
45 }
```

그림 5. 도착 지연시간에 대한 결정트리 구축 처리화면

Fig. 5. Snapshot for Constructing a Decision Tree for the Flight Arrival Delay Time.

그림 4는 항공기 연착 예측 시스템의 빅 데이터 처리를 위한 Hadoop을 기반으로 하는 맵-리듀스 처리과정의

일부를 보여준다. 이 과정을 통해 각 세 부류에서 빅 데이터 내의 관심정보들의 발생빈도에 따라 항공기 연착에 영향을 주는 결정요인에 대한 결정트리를 구성한다. 그림 5는 항공기 운항 빅 데이터에서 비행기 도착 지연 정보와 관련된 결정요인에 따른 의사결정트리를 구축하기 위한 처리과정의 일부를 보여준다.

V. 구현 결과 및 고찰

이 장에서는 빅 데이터를 이용한 항공기 연착 예측 시스템에 대한 구현 및 실행결과를 살펴본다. 그림 6은 결정트리 구축을 위해 하둡 시스템을 이용하여 항공기 운항 빅 데이터를 결정요인 별로 처리한 결과 화면을 보여준다. 또한 각 결정트리에 대한 항공기 연착과 관련된 회귀분석을 한 결과 비행 운항 내부정보인 항공기 기종 수령과의 상관관계는 양의 방향으로 설명되었으며, 비행 운항 외부정보인 비행 조건정보는 음의 방향으로 설명되었고, 마지막으로 항공기 이용 상황정보인 운항 시즌정보와의 상관관계는 양의 방향으로 설명되었다.

	Counter	Map	Reduce	Total
Map-Reduce Framework	Spilled Records	32,964,587	21,900,551	54,865,138
	Map output materialized bytes	333,927,583	0	333,927,583
	Reduce input records	0	21,900,551	21,900,551
	Virtual memory (bytes) snapshot	199,957,258,240	2,286,641,152	202,243,899,392
	Map input records	59,285,466	0	59,285,466
	SPLIT_RAW_BYTES	9,768	0	9,768
	Map output bytes	290,125,953	0	290,125,953
	Reduce shuffle bytes	0	333,927,583	333,927,583
	Physical memory (bytes) snapshot	16,433,020,928	281,681,920	16,714,702,848
	Reduce input groups	0	108	108
	Combine output records	0	0	0
	Reduce output records	0	108	108
	Map output records	21,900,551	0	21,900,551
	Combine input records	0	0	0
	CPU time spent (ms)	193,680	19,240	212,920
	Total committed heap usage (bytes)	13,973,848,064	191,168,512	14,165,016,576
File Input Format Counters	Bytes Read	5,733,396,812	0	5,733,396,812
FileSystemCounters	HDFS_BYTES_READ	5,733,406,580	0	5,733,406,580
	FILE_BYTES_WRITTEN	504,016,841	333,948,878	837,965,719
	FILE_BYTES_READ	168,168,140	333,927,079	502,095,219
	HDFS_BYTES_WRITTEN	0	1,754	1,754

그림 6. 결정트리 구축을 위한 하둡 시스템 처리 결과화면
Fig. 6. Snapshot of the Hadoop System for Constructing a Decision Tree.

그림 7은 하둡 시스템에 구축된 과거의 항공기 연착을 유발했던 빅 데이터 자료를 기반으로 하는 결정트리를 이용하여 특정 연도의 각 월별 항공기 도착 지연과 출발 지연 예측 가능한 시간을 질의하여 그 실행 결과를 보여주는 화면이다. 그림에서 보인바와 같이 2008년 월별 항

```
hive> SELECT Year, Month, AVG(ArrDelay) AS avg_arrive_delay_time,
    >        AVG(DepDelay) AS avg_departure_delay_time
    > FROM airline_delay
    > WHERE delayYear = 2008
    > AND ArrDelay > 0
    > GROUP BY Year, Month;
Total MapReduce jobs = 1
Launching Job 1 out of 1
Number of reduce tasks not specified. Estimated from input data size: 1
In order to change the average load for a reducer (in bytes):
  set hive.exec.reducers.bytes.per.reducer=<number>
In order to limit the maximum number of reducers:
  set hive.exec.reducers.max=<number>
In order to set a constant number of reducers:
  set mapred.reduce.tasks=<number>
Starting Job = job_201508192304_0003, Tracking URL = http://localhost:50030/jobdetails.jsp
?jobid=job_201508192304_0003
Kill Command = /home/hadoop/hadoop/libexec/../bin/hadoop job  -Dmapred.job.tracker=localho
st:9001 -kill job_201508192304_0003
Hadoop job information for Stage-1: number of mappers: 3; number of reducers: 1
2015-08-21 02:51:40,844 Stage-1 map = 0%,  reduce = 0%

2015-08-21 03:11:54,391 Stage-1 map = 100%,  reduce = 100%, Cumulative CPU 12.09 sec
MapReduce Total cumulative CPU time: 12 seconds 90 msec
Ended Job = job_201508192304_0004
MapReduce Jobs Launched:
Job 0: Map: 3  Reduce: 1   Accumulative CPU: 12.09 sec   HDFS Read: 689455525 HDFS Write:
611 SUCESS
Total MapReduce CPU Time Spent: 12 seconds 90 msec
OK
2008    1       32.9789784093877846     26.998343037716475      279427
2008    2       35.2220242235624        29.100845458261325      278902
2008    3       33.05309686443325       27.787201754505087      294556
2008    4       28.466948021019594      22.537276198358724      256142
2008    5       27.727387669678374      21.854393673455764      254673
2008    6       36.225886034667474      29.810112978502655      295897
2008    7       36.050122812984164      30.916041265162684      264630
2008    8       32.683845213713364      28.069522017877926      239737
2008    9       24.856736036338177      19.634876646720684      169959
2008    10      22.739309959413494      17.703783595341594      183582
2008    11      27.186478683900264      21.300386764073917      181506
2008    12      40.27475551974559       33.933164107482185      280493
Time taken: 49.338 seconds
hive>
```

그림 7. 하둡 시스템에서 항공기 연착 질의와 처리 결과 화면
Fig. 7. Snapshot of the Query Process and Results for the Delayed Flights in Hadoop System.

공기 도착과 출발 지연 가능 시간과 실제 연착을 유발 했던 지연 빈도수를 보여주고 있다. 이러한 질의는 특정 지역의 공항 단위로도 가능하며 특정 항공편의 연착시간을 예측하는데 활용된다. 하지만 항공기의 연착이 발생했을 때의 지연 시간에 대한 정확한 예측은 아직 도전과제이며 각 결정요인들에 대한 가중치가 정확한 연착시간 예측에 큰 영향을 미치는 것으로 알려졌다. 따라서 많은 모의실험 등을 통해 이들 결정요인들에 대한 가중치 값을 결정하는 과정이 필요하다고 생각된다.

VI. 결 론

빅 데이터를 이용한 응용들은 최근 매우 다양하게 이용되고 있으나 대부분 방대한 양의 빅 데이터로부터 빈도수가 높은 키워드를 가려내서 이를 이용한 최근 경향이나 속성을 판단하는 응용에 그치고 있는 실정이다. 빅 데이터 처리를 위한 분석 도구로는 단순하게 특정 키워드를 인지하고 그 발생 빈도를 카운터를 이용하여 계산하는 처리 수준에 머물러 있다. 보다 정확하고 정밀한 데이터 분석이나 의사결정을 필요로 하는 스마트 응용을

위해서는 이러한 수준의 통계 처리 기법만 가지고는 좋은 결과를 기대하기 힘들다. 따라서 보다 정확하고 유연한 데이터 분석과 의사결정을 가능하게 하기 위해선 여러 요인 결정에 영향을 미칠 수 있는 결정요인을 복합적으로 반영할 수 있는 의사결정구조를 갖춘 데이터 분석 모델이 필요하다. 본 논문에서는 이러한 복합적인 의사결정모델을 기반으로 하는 스마트 응용을 제안하고 설계하였다.

본 논문에서 스마트 응용으로 빅 데이터를 이용한 항공기 연착 예측 시스템을 설계하였고 복합적인 의사결정 기법으로 분석하였다. 최근 항공기를 이용한 국내외 여행객의 수가 급속히 늘어나면서 크고 작은 비행기 이착륙 지연이 빈번하게 발생하고 있는 상황이다. 이러한 항공기 연착으로 인한 피해를 최소화 할 수 있도록 과거의 빅 데이터 비행 운항 자료를 기반으로 우리가 필요로 하는 유용한 정보를 효율적으로 결정하고 추출해 낼 수 있는 빅 데이터 분석방법을 제안하고 설계하였다.

본 논문에서는 과거의 비행 운항 기록을 기반으로 항공기의 연착을 유발했던 결정요인들을 찾아내어 세 부류의 정보로 나누어 의사결정 트리를 구성하고 이들 결정 트리 간 상호 관련성을 회귀분석을 통해 정의하여 현재 운항중인 항공기에서 과거와 유사한 패턴의 상황이 발생했을 때 과거의 결정요인들로부터 항공기 연착을 유발시킬 수 있는 가능성을 결정하고 이러한 상황에서 지연 가능시간을 예측하여 탑승객과 관계자에게 알려주는 시스템을 설계하고 구현하였다. 이때 과거의 비행 운항 빅 데이터를 세 부류로 나누어 복합 의사결정 모델을 구축하고 각각의 결정 요인들이 항공기 연착에 영향을 미칠 수 있는 정도에 따라 가중치를 두어 보다 정확한 연착 시간 예측을 가능하게 하였다. 향후 연구과제로는 각 결정요인이 항공기 연착에 미치는 정도를 나타내는 가중치를 모의실험을 통해 정하고 실제 항공기 운항 데이터를 이용해 항공기 연착 예측에 적용하는 것이다.

References

[1] John Gantz, David Reinsel, "The Digital Universe 2020: Big data, Bigger Digital Shadows, and Biggest Growth in the Far East," IDCIVIEW, pp. 1-16, 2012.

[2] S. Kim et. al., "A Big Data Application for Anomaly Detection in VANETs," JIIBC, Vol. 14, No. 6, pp. 175-181, 2014.

[3] Gartner, "High-Tech Tuesday Webinar; Big Data Oppourtunities in Vertical Industries," Gartner Big Data Analytics, 2012.

[4] McKinsey, "Big data: The next frontier for innovation, competition, and productivity," Mckinsey Global Institute, 2011.

[5] H. Y. Yang, "Technology Planning Methodology using Big Data," KISTEP, Vol. 12, pp. 3-33, 2012.

[6] Christian Bizer, "The Meaningful Use of Big Data: Four Perspectives-Four Challenges," SIGMOD Record, Vol. 40, No. 4, pp. 56-60, 2011.

[7] M. Y. Lee, "Big Data Technology Trends for Big Data Analysis," J. of Information Processing Systems, Vol. 19, No. 2, pp. 20-28, 2012.

[8] H. Wickham, "The Split-Apply-Combine Strategy for Big Data Analysis," Journal of Statistical Software, Vol. 40, No. 1, pp. 1-29, 2011.

[9] M. Spiliopoulou, "Web Usage Mining for Web site Evaluation," Communication of the ACM, Vol. 43, No. 8, pp. 127-134, 2000.

[10] K. M. Park, H. Park, H. G. Kim, H. Ko, "Review Mining using Lexical Knowledge and Modality Analysis," Proceedings of the 5th International Universal Communication Symposium, 2011.

[11] S. S. Kim et. al., "Neuro-Fuzzy Modeling using Hierarchical Clustering & Gaussian Mixture Model," J. of Fuzzy Logic & Intelligent Systems, Vol 13, No. 5, pp. 512-519, 2003.

[12] D. W. Choi et. al., "Smartphone based Bio Cognition Data Mining Algorithm," Proceedings of 2010 Conference in KOEN, pp. 155-158, 2010.

[13] X. Y. Yang, Z. Liu, Y. Fu, "MapReduce as a Programming Model for Association Rules Algorithm on Hadoop," Proceeding of the 3rd International Conference on Information Sciences and Interaction Sciences (ICIS), pp. 99-102, 2010.

[14] K. Lee et. al., "Semantic Social Network Analysis," J. of Information Processing Systems, Vol. 18, No.

6, pp. 54-67, 2011.

[15] F. Gordon, O. Corby, M. Buffa, "Semantic Social Network Analysis," Proceeding of WebSci'09, Society On-line, 2009.

저자 소개

오 선 진(종신회원)

- 제 6권 제2호 참조
- 현재 세명대학교 정보통신학부 교수

<주관심분야 : 스마트 응용, 그린 IT, 빅데이터, 모바일컴퓨팅, USN 등>

※ 이 논문은 2014학년도 세명대학교 교내학술연구비 지원에 의해 수행된 연구임