한국빅데이터학회지
제4권 제2호, 2019, pp. 207-220

교통 빅데이터의 효율적 저장 및 검색 기술의 설계와 구현

Design and Implementation of Efficient Storage and Retrieval Technology of Traffic Big Data

김기수[1]·이재진[1]·김홍회[1]·장유림[2]·함유근[3]†

디토닉 주식회사[1], 한국교통안전공단[2], 건국대학교[3]

요 약

최근 정보통신기술의 발달은 센서를 바탕으로 수많은 데이터를 구축하고 이를 이용하여 실시간 서비스를 제공할 수 있게 한다. 교통안전공단에서는 디지털 운행기록계를 통해 전국의 상용차의 운행 정보를 수집하고 있다. 전국 상용자의 운행 정보는 교통 분야에서 다방면으로 활용이 가능하다. 그 중 특히 자율주행 분야에서는 실시간으로 운행정보를 분석하여 위험 운전에 대응을 하거나 방지하는데 도움을 줄 수 있다. 그러나 전통적인 데이터베이스 시스템을 이용하여 대용량의 데이터를 실시간 서비스에 적합한 수준의 성능으로 처리하는 데는 한계가 존재한다. 특히 국내에서는 이와 같은 기술적인 문제로 상용차 운행정보의 실시간 분석을 위한 대규모 교통 빅데이터의 처리가 이전에 시도된 적이 없다. 이런 문제를 해결하기 위해 본 연구에서는 새로운 방식의 데이터베이스 서버 시스템 최적화를 진행하였고 실시간 서비스가 가능한 수준임을 확인하였다. 구축된 데이터베이스 시스템을 이용하여 디지털 트윈, 자율주행환경을 마련하기 위한 기반 데이터를 확보할 수 있을 것으로 기대된다.

■ 중심어 : | 교통빅데이터, 디지털운행기록계, HBase, OLTP, 하둡 에코시스템 |

Abstract

Recent developments in information and communication technology has enabled the deployment of sensor based data to provide real-time services. In Korea, The Korea Transportation Safety Authority is collecting driving information of all commercial vehicles through a fitted digital tachograph (DTG). This information gathered using DTG can be utilized in various ways in the field of transportation. Notably in autonomous driving, the real-time analysis of this information can be used to prevent or respond to dangerous driving behavior. However, there is a limit to processing a large amount of data at a level suitable for real-time services using a traditional database system. In particular, due to a such technical problem, the processing of large quantity of traffic big data for real-time commercial vehicle operation information analysis has never been attempted in Korea. In order to solve this problem, this study optimized the new database server system and confirmed that a real-time service is possible. It is expected that the constructed database system will be used to secure base data needed to establish digital twin and autonomous driving environments.

■ Keyword : Traffic Big Data, DTG(Digital Tacho Graph), HBase, OLTP, Hadoop Ecosystem

2019년 12월 06일 접수; 2019년 12월 10일 수정본 접수; 2019년 12월 31일 게재 확정.

† 교신저자 (ykhahm@konkuk.ac.kr)

Ⅰ. 서 론

2016년 세계경제포럼에서 클라우드 슈밥 의장은 4차 산업혁명의 개념을 주창하였다[15]. 4차 산업혁명이란 센서를 통해 수집된 자료를 통해 현실정보를 빠르게 습득하고, 이를 기반으로 의사결정 및 미래예측까지 가능하게 하는 일련의 정보 시스템이 도입되어 다양한 산업이 연결되고 인간의 행동양식과 산업 등 환경에 변화를 불러올 거대한 흐름이다. 4차 산업혁명의 흐름과 함께 사물인터넷과 인공지능 기술이 발달하면서 자율주행 자동차의 보급도 가까워지고 있다.

자율주행 자동차는 영상센서, 적외선 감지센서 등 다양한 센서를 통해 도로상황, 노면상태, 차간거리와 같은 주변의 많은 정보를 파악하고 실시간으로 의사결정이 이루어져야 주행이 가능하다. 뿐만 아니라 자율주행중 AI가 주변의 위험요소를 인지하기 위해서는 실제 사건사고가 발생할 수 있는 위험 운전 행동 등에 대한 학습이 필요하다. 이에 위험 운전 행동과 사건사고 관계에 대한 연구가 활발히 진행되고 있다.[4, 6] 대부분의 연구는 디지털운행기록계와 같은 교통데이터를 기반으로 하고 있으며, 교통정보의 실시간 분석 및 의사결정을 위해서는 수집되는 데이터의 저장 및 검색이 가능해야 한다. 이를 위해 데이터베이스는 방대한 데이터를 저장할 수 있는 유연한 저장공간을 필요로 한다. 또한 다양한 사용자의 데이터베이스 동시 접근 및 초 단위의 응답시간을 확보해야 한다.

노면에 설치된 전자 검지기 및 지능형 검지기, CCTV를 비롯한 영상센서에서 수집되는 다양한 교통정보는 실시간으로 수집되고, 센서 및 감지기술의 발달에 따라 시간이 지날수록 데이터의 양과 종류가 늘어나고 있으며, 축적된 데이터는 도시계획, 교통운영, 안전 등 다양한 활용가치를 가지고 있다. 차량사진, 차량유형을 포함한 교통 HD 비디오 데이터, 속도, 차선번호, 이동방향등의 차량정보, 운영데이터 등 다양한 데이터를 교통 빅데이터로 분류할 수 있다. 본 연구에서는 상용차에 부착된 디지털운행기록계(Digital Tachograph)에서 취득되는 교통정보를 분석의 대상으로 하였다.

아파치의 하둡(Hadoop)은 대량의 데이터를 처리할 수 있는 프레임워크로, '하둡 에코시스템'이라 불리는 다양한 연계 시스템들을 이용하여 손쉬운 저장소 확장, 분산 처리를 통한 빠른 데이터 처리 속도를 보유하고 있어 빅데이터 처리에 주로 사용된다.[13,19]

따라서 연구의 목적은 하둡 에코시스템을 이용하여 실시간으로 축적되는 DTG 데이터를 저장 및 검색할 수 있는 시스템을 구현하고 성능검사를 통해 효율적 분석이 가능함을 확인하는 것으로, DTG 데이터를 탑재하여 분석 서비스의 가능여부를 평가하는 것을 연구의 목표로 한다.

II. 기술적 배경

2.1 디지털운행기록계 (DTG: Digital TachoGraph)

세계적으로 사업용자동차의 1만 대당 교통사고 사망자수는 비사업용 자동차 1만 대당 교통사고 사망자수보다 높게 나타나고 있다. 유럽연합과 일본 등 교통선진국에서는 이러한 문제를 해결하기 위하여 사업용자동차의 운행정보를 기록하고 공공기관에서 관리하도록 하였다. 기록된 운행정보를 사고분석 등에 사용하여 사고율이 감소하는 것을 확인하였으며[9] 운전자에게 정보를 공유하고 운전자가 안전운전을 위해 노력하도록 하여 사고 발생 저하 효과를 확인하였다[16,17]. 국내에서는 2009년 12월 교통안전법을 개정하여 2014년부터 모든 사업용 차량의 DTG 장착이 의무화되었다[5]. 장착된 DTG를 이용하여 택시[3], 화물차[7], 고속버스[2], 운행 정보 분석에 활용한

결과 통행패턴, 졸음운전 위험구간, 위험운전 행태 등을 확인할 수 있었으나, 기술적인 문제로 전체 DTG데이터를 이용한 분석은 없었다.

현재 한국교통안전공단에서는 DTG를 통해 수집된 운행기록 자료를 분석하여 과학적이고 실증적인 운전자 안전관리를 수행하기 위해 운행기록분석시스템(eTAS)을 구축하여 운영하고 있다. 운행기록분석시스템에서는 위험운전 행동기준을 이용하여 운행기록데이터를 가공한 위험운전 통계데이터를 생산하고 있으며 이를 이용하여 상업용 차량의 운전자 안전교육, 위험운전 행동분석, 사고지점 중첩분석 등을 수행하고 있다. 위험운전 행동분석과 사고지점 중첩분석 등은 특정 기간의 운행기록을 기준으로 이루어지는 정적인 데이터로서 교통정책 수립 등 거시적인 관점에서 사용할 수 있는 분석이다.

DTG 데이터는 자동차의 순간속도, 분당 엔진회전수, 브레이크 신호, GPS, 방위각, 가속도 등으로 이루어져 있으며 세부 제원은 <표 1>과 같다.

상업용 차량은 차량의 시동이 켜진 순간부터 시동이 꺼진 순간까지를 하나의 trip으로 정의하여 trip key를 생성하고, 초 단위로 각각의 데이터를 수집하여 저장한다. 모든 데이터는 초 단위로 생성되며 데이터 생성시간은 0.01초까지 저장된다. GPS

〈표 1〉 DTG 데이터 구조

Content	digit	Description	Example
Trip Key	27	Key of trip	C-225190649181121 21363900
DTG Model	20	Model of DTG	SVMS-150
Chassis number	17	Number of chassis	(secret)
Car type	2	11: city bus 12: Rural Bus 13: town bus 14: Intercity bus 15: Express Bus 16: charter bus 17: Special Passenger Car 21: General Taxi 22: Private Taxi 31: General Lorry 32: individual lorry 41: non-business car	31
Car registration number	12	Car registration number	(secret)
Business number	10	Business number of company	(secret)
Driver code	18	number of Driver license	0000000
Trip distance	4	Trip distance(km), 0000~9999	148
Cumulative trip distance	7	Cumulative trip distance(km), 0000000~9999999	369512
Velocity	3	Velocity(km/h), 000~255	0
RPM	5	Revolution per minute, 0000~9999	630

Break	1	0(off) or 1(on)	0
GPS X	11	Decimal(ex:127.123456*1000000=>127123456)	126967973
GPS Y	11	Decimal(ex:127.123456*1000000=>127123456)	37034581
Azimuth	3	0~360	140
Acceleration X	5	-100.0~100.0	0
Acceleration Y	5	-100.0~100.0	-1.2
Communication status	2	00: Driving recorder normal 11: Position tracking device (GPS Receiver) error 12: Speed sensor error 13: RPM sensor error 14: Brake signal sensor error 21: Sensor input unit error 22: Sensor output unit error 31: Data output unit error 32: Communication device error 41: Distance calculation error 99: Power supply error	00
Service area	2		41
Transportation company location	2		00
Transportaton company	7		(secret)
Date of occurrence of information	14	YYMMDDhhmmssss	18112121363900

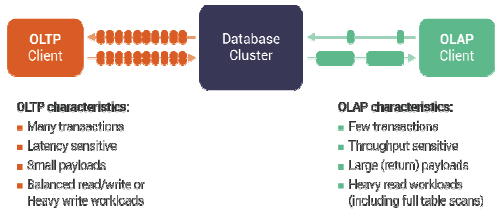
좌표를 통해 차량의 위치를 확인할 수 있고 방위각으로 진행 방향을, RPM, 속도, 브레이크, 가속도를 통해 현재 주행상황을 확인할 수 있다. 운전자코드, 일 주행거리를 통해 운전자 피로도와 차량 피로도를 확인할 수 있고 총주행거리는 차량의 노후화를 식별할 수 있는 지표가 된다.

축적된 DTG 데이터는 차량 유형, 운행지역, 계절, 노후화 정도 등 다양한 기준에 따라 분석될 수 있으며, 실시간으로 전송된 데이터를 저장하여 분석에 사용할 수 있다면 실시간 단위의 전국 상업용 차량의 분포 및 교통량 추정, 도로상황, 교통수요 등을 파악할 수 있다. [3,14] 이러한 데이터는 자율주행차량 운행을 위해 필요한 기초정보로 사용될 수 있으며, 스마트시티의 실시간 도로관리와 신규 교통정책 수립에 도움을 줄 수 있다.

2.2 교통 OLTP용 데이터베이스 시스템

데이터베이스 시스템을 설계하기 위해서는 시스템의 목적과 데이터의 특성을 고려하여 OLTP (On-Line Transaction Processing)와 OLAP (On-Line Analytical Processing) 중 어떤 형태의 서비스를 제공할 것인지 결정하여야 한다. <그림 1>과 같이 OLAP는 많은 양의 데이터를 검색하고 생성하는 더 긴 작업을 의미하며, OLTP는 적은 양의 데이터로 많은 수의 빠른 트랜잭션을 특징으로 한다. 따라서 OLAP는 처리량 중심으로 DB를 평가하고, OLTP는 대기시간 중심으로 DB를 평가한다[27].

RDBMS는 데이터를 안전하게 저장하는 영속성에 중점을 두었기 때문에, 응답시간이 길고 데이터베이스에 동시에 접근하는 것이 불가능하여



〈그림 1〉 OLTP와 OLAP의 일반적인 특성[27]

빠른 속도로 대량의 데이터를 검색하기 어려웠다. 반면 실시간 처리를 할 수 있는 실시간 데이터베이스 시스템에서는 시간제약, 중요도, 가치함수와 같은 정보를 이용하여 여러 사용자가 동시에, 빠른 응답 시간 안에 데이터를 조회하는 것에 가치를 두었기 때문에 실시간 트랜잭션을 스케줄링하고, 시간의 제약을 만족하며 데이터의 일관성을 유지하여야 한다[1]. 실시간 데이터베이스 시스템은 OLAP 기준과는 다르게 OLTP 기준으로 설계되어야 한다. 이는 데이터를 저장하고 검색하는 순간에도 새로운 데이터가 생성되어 전송되고 있으므로 실시간 데이터베이스 시스템인 OLTP에 중점을 두고 장애 허용량과 처리량이 높은 하둡 빅데이터 기술을 적용하여 설계되어야 하기 때문이다. 따라서 OLTP에 적합한 실시간 데이터베이스 시스템을 구축하기 위해 본 연구에서는 기존에 통상적으로 사용하던 관계형 데이터베이스 대신 비관계형 데이터베이스를 선택했으며 이는 비관계형 데이터베이스의 경우, 관계형 데이터베이스에 비해 스키마에 대한 제약이 없어 데이터처리가 유연하고 추후 데이터양 증가에 따라 확장이 용이한 장점이 있기 때문이다. 또한, 하둡 에코시스템을 이용하여 분산처리 환경을 구축했을 때 데이터베이스에 대한 대량의 동시접속이 가능하고 보관하는 데이터량이 증가하더라도 인덱싱 정책에 따라 검색 성능의 저하를 최소화할 수 있다.

2.3 하둡 에코시스템(Hadoop Ecosystem)

K. Zvarevashe et al[11]는 '전 세계 데이터의 90%가 지난 2년간 생성되었다'고 했는데 이와 같이 실시간으로 생성되는 방대한 양의 데이터를 다루기 위해서는 분산 처리가 필수적이다. 하둡은 클러스터 컴퓨터 환경에서 동작하는 분산 응용 프로그램을 지원하는 프레임워크로 대량의 데이터를 처리할 수 있는 대표적인 빅데이터 기술이다[10]. 하둡을 기반으로 연계해서 사용할 수 있는 다양한 주변 시스템들을 '하둡 에코시스템'이라 한다. 하둡 에코 시스템은 대표적으로 분산처리를 위한 MapReduce와 분산 저장을 위한 HDFS가 있다. 본 연구에서는 하둡 에코 시스템에서 저장을 담당하는 HDFS, HBase와 분석을 위한 Zeppelin, 데이터 흐름 관리 엔진인 NiFi, 마지막으로 분산처리를 담당하는 Spark을 활용하여 빅데이터 시스템을 구축하였다.

HDFS(Hadoop Distributed File System)[22]는 HDFS에 저장된 파일들의 메타 정보를 관리하는 네임 노드와 실제 파일이 저장되는 데이터 노드로 구성된 마스터-슬레이브 기반 분산 파일 시스템이다. 파일을 저장하면 특정 사이즈의 블록으로 나뉘어 분산된 서버에 저장되고, 저장된 블록은 복제되어 다른 데이터 노드에 분산 저장된다. HDFS에 저장된 파일을 읽기 위해서는 사용자가 네임노드를 통해 원하는 파일이 저장된 블록의 위치를 확인하고, 해당 블록이 저장된 데이터 노드에서 직접 데이터를 조회한다[12].

HBase[23]는 하둡 에코 시스템에서 데이터 분산 저장 및 조회를 위해 사용하는 비관계형 데이터베이스 중의 하나다. 구글의 Bigtable을 모델로 하여 개발된 분산 컬럼 기반의 데이터베이스로 HDFS와 같이 마스터-슬레이브 구조를 갖는다. HBase는 분산시스템을 통제하는 마스터에 의해 복제된 HDFS상의 데이터 파일에 대한 일관성과 최신성을 보장한다. 또한 하둡 에코시스템에 속

하는 처리 솔루션과 연동하여 데이터 처리의 소스와 처리 결과에 대한 목적지로 유연하게 사용할 수 있으며 스키마에 대한 제약이 없다. HBase의 검색 효율은 각 Row마다 존재하는 Rowkey에 따라 달라진다. HBase에서는 Rowkey -> Column Family -> Column Qualifier -> Time stamp의 순서로 사전적 정렬을 적용하므로 데이터 따른 효율적인 저장 전략을 수립하는 것이 필요하다.

NiFi[26]는 NSA에서 Apache에 기증한 오픈소스 데이터 플로우 엔진으로 시스템 간 데이터 흐름을 자동화 하도록 설계된 소프트웨어이다. NiFi에는 HTTP와 같은 네트워크 통신용에서부터 데이터의 변환, 시스템 내 명령 호출, 하둡 빅데이터 에코시스템 연동까지 다양한 기능의 프로세서들을 제공한다. NiFi는 이러한 다양한 프로세서의 동작 옵션 설정 및 프로세서간의 흐름 설계를 위한 웹 기반의 사용자인터페이스를 제공하고 있으며 이를 이용하여 손쉽게 외부 시스템과의 연동 인터페이스를 제공할 수 있다.

Zeppelin[25]은 웹 기반 분석 프레임워크로 Spark을 활용한 분산 처리부터 시각화, R과 Python 기반 분석 환경, HBase등 다양한 하둡 분산 소에 대한 접근 등 분산 환경에서 데이터 분석이 가능하도록 다양한 기능들을 제공한다. 본 연구에서는 이러한 Zeppelin의 기능들을 활용할 수 있는 분석 환경을 제공하였다.

Apache Spark[24]은 메모리 기반 분산 처리 엔진으로 RDD(Resilient Distributed DataSet)이라는 분산 데이터 객체를 제공하여, 분산 환경에서 데이터 처리가 가능하도록 하였다. 뿐만 아니라, DataFrame, DataSet과 같은 데이터 구조를 지원하여 조회, 통계, 분석의 역할을 지원하고, Spark ML을 통한 머신러닝 분석도 지원한다.

2.4 교통정보시스템 사례

청주시 교통 빅데이터 분석 시스템[8]은 관계형 데이터베이스 시스템 기반의 배치 처리방식으로 구축되었다. 하루 1회 버스정보시스템(Bus Information System, BIS), 첨단교통관리시스템(Advanced Traffic Management System, ATMS) 데이터를 첨단교통정보시스템(Advanced Traffic Management System, ATIS) DB에 수집한다. 수집된 원시 데이터는 DBMS에서 ATIS DW로 변환되어 하루 1회 데이터를 이전한다. 복잡한 교통문제 연구를 위한 기초적인 자료 제공을 목적으로 시스템을 구축하였으나 배치 처리 방식을 사용하고 있어 빅데이터 기술을 도입할 것을 향후 추가적인 연구 방향으로 설정하였다.

청주시 사례와 같이 기존의 관계형 데이터베이스 시스템으로 구축된 교통 데이터는 축적된 데이터의 규모가 증가할 때 스토리지를 확장하기가 어려웠기 때문에, 데이터 품질 증가로 개별 데이터의 용량이 증가하면 보관중인 데이터의 갱신 주기가 짧아져 장기간의 축적된 데이터를 분석하기가 어려웠다.

Intel[27]은 중국 저장성의 도시를 대상으로 하둡 기반의 빅데이터 처리 시스템을 운영한 사례를 소개하였다. 이 도시는 생산된 데이터가 서로 다른 데이터 센터에 저장되어 있어 활용이 어려웠고 데이터의 규모로 인해 저장 기간이 감소하고 있었는데, 분산 처리시스템을 이용하여 중앙 집중식 관리, 방대한 데이터 활용 최적화, 도시 전역의 교통류 개선이라는 효과를 거두었다. S. Amini, I. Gerostathopoulos and C. Prehofer[14]는 실시간 교통제어를 목적으로 카프카를 사용하여 포괄적이고 유연한 아키텍처를 제안하였고 제작한 프로토타입 플랫폼에서 시뮬레이션을 수행하여 고속도로 차선관리에 기여할 수 있음을 확인하였다.

Zeng, G.[20]는 지능형 교통제어 시스템에 하둡 에코시스템 기반의 빅데이터 처리 기술을 적용하여 검지기의 교통류, 평균속도, 혼잡도 등을 식별하였다.

위 3 가지 사례와 같이 빅데이터 기술이 발전하면서 등장한 하둡과 같은 분산 처리 시스템을 이용하여 시스템을 구축하면 기존 관계형 데이터베이스 대비 저장소의 용량 확장이 간편하며 비정형 데이터의 저장 및 통합관제 측면에서 이점이 있다. 또한, 디지털 운행기록계에서 수집되는 운행기록정보는 초 단위로 축적되는 정보로 실시간 서비스를 제공할 수 있는 활용가치가 있으나 기존 시스템에서는 빠른 속도로 축적되는 데이터를 저장하는 동시에 검색하기가 어려웠다. 분산처리 기술을 활용하여 실시간으로 축적되는 데이터의 저장 및 검색 시스템을 구축할 수 있을 것으로 판단하였다.

Ⅲ. OLTP용 교통 빅데이터 데이터베이스 시스템 설계 및 제안

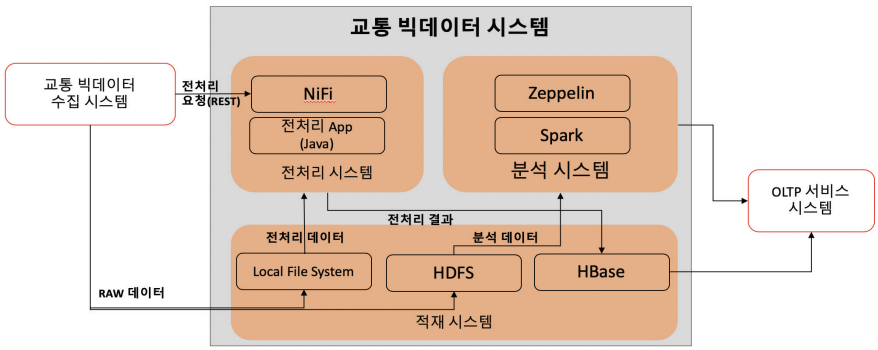
3.1 제안하는 빅데이터 시스템 구조

본 연구에서 제안하는 시스템은 <그림 2>와 같다. 빅데이터 시스템은 그 역할에 따라 다시 전처리, 분석, 적재 시스템으로 분류할 수 있다. 교통 빅데이터 시스템 외에 수집 시스템과 OLTP서비스 시스템이 있지만, 본 연구에서는 OLTP 서비스를 위한 데이터 처리 시스템을 제안하는 것이므로 OLTP 시스템 부분은 논의하지 않는다.

전처리 시스템은 별도의 수집 시스템에서 수집한 데이터를 전달받아 비정형 데이터베이스에 저장하는 역할을 수행한다. 전처리 시스템은 크게 NiFi와 별도의 가공 프로세스가 포함된 자바 어플리케이션으로 구성되어 있다. NiFi는 주로 수집 시스템과의 인터페이스의 역할을 수행하며, 전처리 과정은 자바 어플리케이션에서 담당한다. 이와 같이 전처리 시스템을 구성한 이유는 다양한 형태의 데이터를 수집하고 처리하려면 해당 데이터에 맞는 전처리 방식이 필요하기 때문이다. 따라서 이와 같이 전처리 시스템을 구성한다면, 데이터에 따라 달라지는 전처리 과정을 하나의 시스템에서 관리할 수 있게 된다.

적재 시스템은 수집 시스템을 통해 전달된 데이터를 저장하고 관리한다. 적재 시스템은 임시 데이터 저장소인 LFS(Local File System)와 1회성 분석 데이터를 저장하는 분산 파일 시스템인 HDFS, 전처리 결과를 저장하고 빠른 조회가 가능한 비정형 데이터베이스인 HBase로 분류할 수 있다.

분석 시스템은 적제 시스템에 저장하고 있는 데이터를 분석에 필요한 데이터를 가공하고, 분석이 가능한 환경을 제공하는 역할을 수행한다. 분석 시스템에서 처리 및 분석을 수행한 데이터는 별도의 백업 과정 없이 바로 서비스 시스템에



〈그림 2〉 교통 빅데이터 시스템 구조

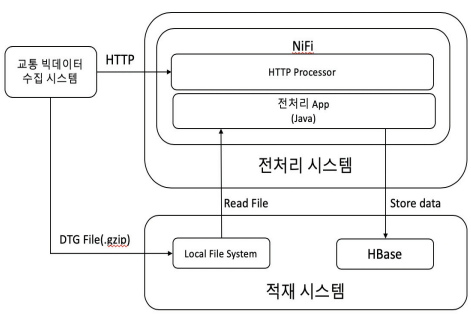
전달된다. 각 시스템에 대한 자세한 내용은 아래에서 설명하도록 한다.

3.2 전처리 시스템

교통 빅데이터에서 수집 시스템은 사용하는 데이터의 형식과 수집 주기, 크기 등에 따라 다양한 구성이 가능하다. 본 연구에서 다루는 DTG데이터는 1일~1달 주기로 데이터를 수집 및 저장한다. 따라서 본 연구에서는 이처럼 다양한 수집 시스템을 지원할 수 있도록 NiFi 기반의 전처리 시스템을 구성하였다.

본 연구에서는 DTG 데이터의 전처리를 위해 <그림 3>과 같이 전처리 시스템을 구성하였다. 먼저 수집 시스템은 압축된 형태로 DTG파일을 NiFi가 설치된 서버의 LFS에 적재한다. 수집 시스템이 압축된 상태로 데이터를 전달하는 이유는 하루 동안 수집된 약 1TB의 텍스트 파일을 물리적으로 다른 공간에 있는 빅데이터 시스템에 전달할 때 발생하는 네트워크 비용을 최소화하기 위함이다. 또한 압축된 파일은 순차적으로 데이터를 읽어서 처리해야 하기 때문에 분산 환경에서 처리가 복잡하다. 따라서 본 연구에서는 압축된 파일을 LFS와 같은 단일 저장소에 적재한다.

LFS로의 적재를 완료하면 수집 시스템은 적재가 완료되었음을 HTTP를 통해 NiFi에 전달한다. 이를 통해 NiFi는 새로운 파일이 LFS에 적재되었음을 확인하고, 전처리 어플리케이션을 구동시킨다. 구동된 전처리 어플리케이션은 해당 파일을 압축된 상태로 읽어 전처리한 다음 HBase에 저장한다. 전처리 과정에서 압축된 파일을 풀지 않고 압축된 상태로 처리하는 이유는 압축된 파일이 텍스트 기반 파일이며, 압축을 풀 경우 처리해야할 데이터의 양이 많아져 처리 시간에 영향을 주기 때문이다.



〈그림 3〉 전처리 시스템 구조

전처리 어플리케이션은 압축된 파일에서 읽은 데이터를 HBase에 저장하는 과정에서 빠른 검색이 가능하도록 HBase에 저장하는 각 데이터에 대해 RowKey를 생성한다. RowKey는 공간, 시간, 차량 정보와 같은 속성을 조합하여 생성하며, 이는 활용하는 OLTP서비스에 따라 다르게 구성할 수 있다.

3.3 적재 시스템

적재 시스템은 데이터의 생명주기, 활용성, 크기, 형태 등 다양한 데이터의 특성에 따라 저장할 저장소를 선정하여야 한다. 예를 들어, 압축된 DTG 데이터 파일은 압축된 상태로 전달 받으며 HBase에 저장하면 재사용할 필요가 없다. 이런 종류의 파일들은 임시적으로 저장소에 적재하고 처리가 된 후에는 시스템에서 삭제한다. 뿐만 아니라 2절에서 언급한 바와 같이, 압축된 파일은 분산 환경에서 처리가 어렵기 때문에 LFS와 같은 단일 저장소가 적합하다. 결과적으로 데이터의 특징에 따라 분산 환경에서 저장할 것인지, 아니면 데이터베이스를 활용할 것인지에 대한 결정이 필요하다. 본 연구에서는 DTG데이터의 특성과 가공 처리된 데이터의 생명주기에 따라 다른 저장소에 적재하여 관리한다.

본 연구에서 사용한 저장소는 크게 LFS와 HDFS, HBase가 있다. LFS는 앞서 전처리 시스템에서 설명한 것과 같이 수집시스템에서 압축된 형태로 전달된 DTG데이터를 임시적으로 관리하는

역할을 한다.

3.4 분석 시스템

교통 빅데이터 분석 시스템을 선정할 때 고려한 사항은 분석 환경과 대용량 DTG데이터에 대한 빠른 처리이다. 먼저 분석 환경은 분석 결과에 대한 시각화, 저장된 데이터의 활용, 모델 처리나 통계 처리 가능 여부, 분산 환경에서 분석 작업 수행 여부 등을 고려하여 하둡 에코시스템 중 하나인 Apache Zeppelin을 사용하였다. 본 연구에서 Spark은 분석 과정에서 직접적인 역할을 하지는 않는다. Spark은 Zeppelin과 연동하여, 분석 시 중간 데이터 가공 및 처리에 필요한 시간을 단축하거나, 연계 서비스에서 필요로 하는 분석 결과를 제공하기위한 개발 및 운영 환경의 역할을 수행한다.

3.5 시스템 튜닝

하둡 에코시스템은 교통뿐만 아니라, 다양한 분야에서의 활용을 목표로 하고 있다. 따라서 분야나 사용하는 데이터에 따라 하둡 에코 시스템을 튜닝하여 사용함으로써 성능에 대한 최적화를 진행한다. 제안된 빅데이터 시스템에서는 HDFS와 HBase의 튜닝을 통해 적재와 조회 성능에 대한 최적화를 진행하였다.

우선 HDFS와 HBase는 마스터 노드와 슬레이브 노드로 구성된 하드웨어의 규모에 따라 퍼포먼스의 표출이 천차만별이다. HDFS의 경우 접근 가능한 스레드 수, 사용할 메모리, 저장공간 할당 등에 대한 다양한 설정을 제공하고 있으나 사용 환경에 따라 설정값의 면밀한 검토를 요한다. HBase에서 보관하는 HFile역시 HDFS상에 보관되므로 기본적인 파일 엑세스와 보관, 처리 성능이 HDFS 설정에 종속된다. HDFS 설정을 최적화 한 후에 HBase에서 사용하는 스레드, 메모리, 저장공간 등의 설정을 서비스 형태에 맞게 설정해야 한다.

HDFS의 튜닝은 크게 Thread와 Network관점에서 진행하였다. HDFS에서 외부 요청을 처리하는데 사용할 Thread 수인 데이터 노드 핸들러 카운터의 기본값은 3으로 설정되어 있으나 20으로 변경하여 실시간으로 처리할 수 있는 Thread 수를 확보하였고, 하나의 데이터 노드에서 동시에 서비스 가능한 HDFS 블록의 개수 제한을 기본값 256에서 1024로 변경하였다. TCP/IP network에서는 TCP 패킷의 overhead를 절약하고자 Nagle's algorithm이 활성화되어 있어 작은 크기의 패킷들을 모아서 전송하도록 설정되어 있는데, 본 연구에서는 OLTP 용도로 HBase를 사용하고자 하였으므로 tcp노딜레이 설정을 활성화함으로써 패킷을 모으지 않고 바로 전송하도록 하여 latency를 상승시켰다.

HBase는 적재, 저장 및 관리, 검색적 측면에서의 최적화를 수행하였다. 먼저 대용량 DTG데이터를 빠르게 적재하기 위해 자바 어플리케이션에 대한 분산 병렬 처리를 지원하도록 구현하였다. 또한 데이터에 따라 사용되는 스레드와 프로세스를 할당할 수 있도록 하여 데이터 크기에 따른 유동적 할당이 가능하도록 구성하였다. 분산 병렬환경에서 자바 어플리케이션은 스레드당 50만 라인 단위로 데이터를 처리 하도록 하였으며, 각 프로세스당 최대 2개의 스레드를 사용할 수 있도록 하였다. 단, 대상 파일이 압축된 상태이기 때문에, 파일을 읽어 각 스레드의 버퍼에 저장하는 과정은 단일 스레드에서 수행해야 한다.

HBase에서 효율적인 저장공간 사용 및 검색 효율 증가를 위해서는 HBase에 저장하는 Row의 컬럼 수를 최소화해야 한다. HBase에서 컬럼은 RowKey이외에 HBase 내의 데이터를 검색할 수 있는 검색어로 사용할 수 있는데, HBase는 서로 다른 컬럼을 저장할 때 동일한 RowKey를 갖는 서로 다른 Row로 컬럼들을 저장한다. 그렇기 때문에 무의미한 컬럼이 많을 경우, HBase 내에서는 Row의 수가 많아져, 불필요하게 저장공간을

사용하거나 검색 성능을 저하시키는 원인이 된다. 본 연구에서 사용한 DTG데이터는 총 20개의 컬럼이 있지만, 조회 시 직접적으로 사용하는 컬럼만 조회 가능하도록 하여 총 2개의 컬럼만 HBase 내에 존재하도록 구성하였다.

HBase에 저장된 DTG데이터의 RowKey는 검색조건을 한정할 때, 한정된 검색조건에서 성능을 확보할 수 있는 형태로 구성하였다. 한국교통안전공단에서는 차량의 운행기록정보를 기반으로 위험운행 행동분석을 수행하여 운전자의 위험운전 여부 및 주기를 식별하고 운전자 교육, 운수회사 관리 및 점검에 사용한다. 위험운행 행동분석을 위해 초 단위 또는 Trip 단위로 DTG 데이터를 추출하므로, 이를 고려하여 초 단위 또는 Trip 단위의 검색이 가능하도록 RowKey를 차대번호와 trip key, 운행시각의 조합으로 구성하였다. 차대번호는 지역번호가 포함되어 hex 분산이 가능하며, trip key는 조회의 기본 단위이고 운행시각은 해당 row 데이터를 식별할 수 있는 고유식별자이다. 동일 차량의 동일 트립 내에는 1초당 1건의 데이터가 존재해야 하고 동일한 데이터가 존재하지 않는 고유 데이터이므로 최소 식별단위로 기능한다.

HBase는 RowKey 사전적 정렬 특성을 가지고 있어 동일차량의 동일 트립 데이터는 운행시각 순으로 정렬되며 조건에 맞는 scan시 점프 scan 없이 연속으로 데이터를 출력하여 최초 반응이 빠르고 단절 없이 데이터를 고속으로 획득할 수 있다.

검색조건은 차량으로만 검색하는 방법, 특정 차량의 특정 트립으로 검색하는 방법, 특정 차량의 특정 트립 내 특정 시간 범위로 검색하는 방법 중 선택할 수 있다. 데이터 수집 과정에서 차종 등의 별도 통계정보를 수집하여 postgresql 등의 메타DB에 기록하고 대상 차량정보만 원활하게 획득할 수 있으면 기존 RDBMS 대비 고속으로 대용량의 데이터를 검색할 수 있을 것으로 판단된다.

3.6 제안된 시스템 검증

본 논문에서는 교통 빅데이터의 효율적인 저장 및 검색이 가능한 데이터베이스 서버 시스템을 구축하기 위해 한국교통안전공단에서 수집하고 있는 전국 상업용 차량 운행자료를 수집하였다. 데이터의 공간적 범위는 DTG를 사용하고 있는 차량이 있는 전국을 대상으로 하며, 시간적 범위는 운행기록분석시스템에서 전처리가 완료된 최신 데이터인 2018년 12월에서 2019년 7월까지의 데이터를 사용하였다.

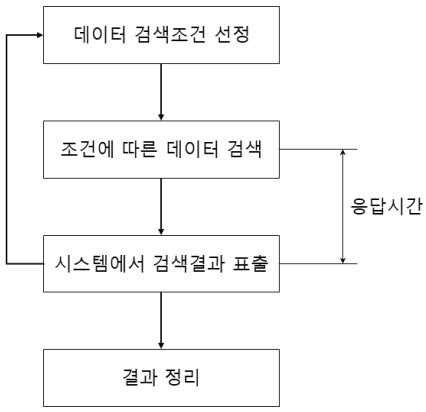
시스템 사양은 <표 2>와 같으며, 클라이언트는 Windows 10을 사용하였다. 실험모델은 분석가가 다양한 데이터를 사용하는 것을 고려하여 서로 다른 운수회사의 데이터가 요청되도록 설정하고, 운행시간은 동일한 값으로 하였다. 즉, 실험 데이터는 동일한 운행시간에서 동일한 크기로 발생된다. 실제 분석 과정에서 분석자가 선택하는 노선 및 차량 종류에 따라 운행시간이 달라지기 때문에 다양한 값을 고려하였다. 실제 분석을 위한 데이터 추출 시에는 동일 노선의 데이터를 추출하여 Trip 시간이 다른 경우도 사용할 수 있으나 실험에서는 동일한 운행시간에서 동작하는 응답시간을 확인하였다.

성능시험 평가의 모식도는 <그림 4>와 같다. 먼저, RowKey 기반으로 데이터 검색조건을 선정한다. 성능시험 평가에서 응답시간에 영향을 줄 수 있는 데이터 크기를 종속변수로 설정하기 위해

〈표 2〉 시스템 사양

분류	사양
네임노드 2기	8코어 cpu(16스레드) 메모리 128GB
데이터노드 그룹1- 13기	8코어 cpu(16스레드) 메모리 32GB
데이터노드 그룹2- 15기	VM 전용 박스에 구성 8스레드 가상 CPU 물리 메모리 32GB 할당
전체 HDFS 용량	약 97TB

*복제 정책 : 원본 포함 3개 유지



〈그림 4〉 성능시험 평가 프로세스

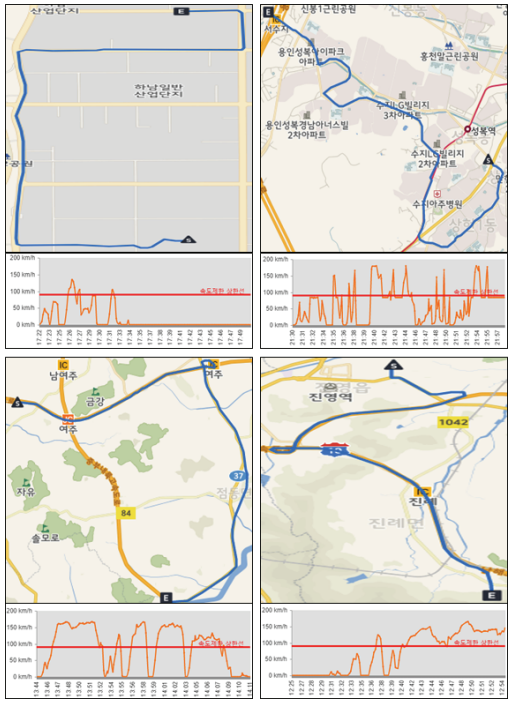
운행시간을 기준으로 검색조건을 설정하였다. 버스, 택시, 화물차의 모든 상용차에서 측정할 수 있도록 운행시간은 30분, 1시간, 1시간 30분, 2시간으로 설정하였다. 데이터베이스 시스템의 초기화 시점에서 설정된 검색조건에 따라 데이터 검색 버튼을 클릭한다. 버튼을 클릭한 시점부터 검색결과가 표출되는 시점까지를 응답시간으로 측정한다. 검색결과는 시스템에서 출발위치와 도착위치, 전체 운행경로를 확인할 수 있는 Map data와 운행시각에 따른 차량의 속도변화 및 속도제한 상한선 그래프로 구성되어 있으며, 모든 결과가 표출되었을 때를 응답종료시점으로 보았다. 동일한 운행시간에 대해 반복해서 응답시간을 측정하고, 검색조건을 변경하여 재측정을 수행하였다.

운행시간 및 시도횟수에 따른 실시간 검색의 응답시간을 다음 표와 그래프로 정리하였다. 운행시간이 길어질수록 전체 데이터의 수가 증가하기 때문에, 운행시간에 따라 응답시간이 증가하는 것을 확인할 수 있었다. OLTP서비스가 가능한 시간은 수 초 이내로, 본 성능 시험 평가에서는 3~5초 사이에 결과가 표출되면 서비스가 가능한 것으로 판단하였다.

<표 3>과 같이, 성능 시험 평가 결과 1시간 단위의 운행데이터를 호출할 때 응답시간은 2초 이내이며, 2시간 단위의 운행데이터는 3초 이내에 호출되어 OLTP서비스를 제공할 수 있는 응답시간 이내에서

〈표 3〉 운행시간에 따른 평균 응답시간

Operating time(h)	0.5	1.0	1.5	2.0
Average of Response time(s)	1.62± 0.06	1.91± 0.04	2.27± 0.16	2.82± 0.28



〈그림 5〉 데이터베이스 시스템의 검색결과 UI

동작함을 확인할 수 있었다.

<그림 5>는 구축된 시스템의 데이터 검색결과 UI를 나타낸 것이다. 검색결과 Trip 단위의 DTG 데이터를 지도에 표출하여 출발위치와 도착위치, 전체 운행경로를 확인할 수 있고, 해당 Trip에서 운행시각에 따른 차량의 속도변화 및 속도제한 상한선을 그래프 형태로 확인할 수 있다. 그래프를 통해 과속, 장기과속, 급가속, 급출발, 급감속, 급정지 등의 위험운전 정보를 식별할 수 있다.

Ⅳ. 결 론

본 논문에서는 대용량의 교통 빅데이터를 저

장하고 실시간 운행정보 분석에 적합한 수준의 검색을 위해서 하둡 에코시스템 기반의 분산 환경에서 데이터베이스 서버 시스템을 구축하였다. 구축된 데이터베이스 서버 시스템은 약 9천억건의 DTG 교통 빅데이터를 분산 저장하고 검색 시 분산처리를 수행한다. 성능시험 결과 2시간 이내의 Trip데이터를 호출할 때 OLTP 서비스에 적합한 3초 이내의 응답시간을 제공하는 것을 확인할 수 있었다. 본 연구에서 구축된 시스템에서 운행정보를 OLTP에 적합한 응답시간 내에 확인할 수 있다면, 향후 교통관리 측면에서 특정 지역의 교통량 추정 및 교통수요를 실시간 단위로 분석하는 것이 가능할 것으로 판단된다. 또한 교통정보는 시공간정보를 모두 포함하는 데이터이므로, 실시간 단위 분석을 통해 향후 자율주행 지원, 스마트 도시 관리 등에 활용할 수 있는 시공간 빅데이터 플랫폼의 개발 및 최적화에도 기여할 수 있을 것으로 판단된다.

연구의 한계는 다음과 같다. 성능시험에서 2시간 단위의 운행데이터 호출까지 3초 이내의 응답시간을 확인하였으나, 장시간 운행하는 화물차의 경우 7시간, 10시간, 18시간 등 매우 오랜 시간의 Trip data가 존재한다. 성능시험 결과에서 데이터의 양이 증가함에 따라 응답시간이 선형적으로 증가하는 것을 확인할 수 있었다. 데이터가 일정 크기를 초과하는 경우에도 OLTP 서비스가 가능하도록 일정 크기를 기준으로 서로 다른 최적화 방법론을 적용하여 시스템을 개선할 필요가 있다. 또한, 다양한 운영체제 환경과 시스템 구성 방법에 의해서 장시간의 시스템 운영과 보완이 이루어져야 한다. 본 시스템은 빅데이터에 대응하기 위한 하둡 에코시스템 기반으로 구축되었으나, 향후 지속적인 데이터의 축적에 따른 전체 DB 크기의 증가로 인하여 서버 시스템의 성능이 저하되는 경우 OLTP 서비스에 적합한 검색 성능이 확보되지 않을 수 있기 때문이다.

이 데이터베이스 서버 시스템을 이용하여 추후 DTG 데이터를 사용한 교통 빅데이터의 분석 작업을 기존 RDBMS 대비 빠르게 수행할 수 있을 것으로 기대된다. 교통 빅데이터 분석결과는 자율주행 환경을 구축하기 위한 기반 데이터로 사용될 수 있을 것이며, 빅데이터의 효율적인 저장 및 검색 기술은 디지털 트윈 등 다른 범주의 데이터를 저장, 관리하기 위한 시스템을 구축할 때에도 사용될 수 있을 것이다.

참 고 문 헌

[1] 권오수, 홍동권, “실시간 검색을 위한 다중 사용자용 주기억장치 자료저장 시스템 개발”, 한국정보처리학회지, 제10D권, 제2호, pp.187-194, 2003.

[2] 김수재, 주재홍, 추상호, 이향숙, “고속버스 DTG 자료를 활용한 버스 위험운전 행태 분석”, 한국ITS학회지, 제17권, 제2호, pp.87-97, 2018.

[3] 안상하, 신용은, “DTG자료 기반 택시 이용자 통행패턴 분석: 부산시 택시 사례”, 대한토목학회 논문지, 제38권, 제6호, pp.907-916, 2018.

[4] 유재곤, 유재영, 김종배, “운전 습관 개선을 위한 위험 운전 분석 어플리케이션의 설계 및 구현”, 한국정보통신학회: 학술대회논문집, pp.301-303, 2015.

[5] 임준범, 유수재, “DTG자료를 활용한 화물 자동차 운전행태 분석과 개선방향 연구”, 대한교통학회지, 제12권, 제5호, pp.28-33, 2015.

[6] 정은비, 오철, 강경표, 강연수, “V2X 환경에서 위험운전이벤트 검지 및 분석을 통한 교통안전 모니터링기법 개발”, 한국ITS학회 논문지, 제11권, 제6호, pp.1-14, 2012.

[7] 조종석, 이현석, 이재영, 김덕녕, “화물차 DTG 데이터를 활용한 고속도로 졸음운전 위험구간 분석”, 대한교통학회지, 제35권, 제2호, pp.160-168, 2017.

[8] 청주시, 청주시 교통정보 빅데이터 분석 시스템

구축, 2015.

[9] De Waard,D. and Rooijers,T. “An experimental study to evaluate the effectiveness of different methods and intensities of law enforcement on driving speed on motorways” *Accident Analysis & Prevention*, Vol.26, No.6, pp.751-765, 1994.

[10] K.Shvachko, H.Kuang, S.Radia, and R.Chansler, “The hadoop distributed file system”, Mass Storage Systems and Technologies (MSST), pp.1-10, 2010.

[11] K.Zvarevashe and T.T.Gotora, “A Random Walk through the Dark Side of NoSQL Databases in Big Data Analytics”, *International Journal of Science and Research*, Vol.3, pp.506-509, 2014.

[12] Konstantin V.Shvachko, “HDFS Scalability: The Limits to Growth”, ;login:: the magazine of *USENIX & SAGE*, Vol.35, No.2, pp.6-16, 2010.

[13] Liao,H., Han,J., and Fang,J., “Multi-dimensional Index on Hadoop Distributed File System”, 2010 IEEE Fifth International Conference on Networking, Architecture, and Storage, 2010.

[14] S. Amini, I. Gerostathopoulos and C. Prehofer, “Big data analytics architecture for real-time traffic control”, 2017 5th IEEE International Conference on Models and Technologies for Intelligent Transportation Systems (MT-ITS), pp. 710-715, 2017.

[15] Schwab, Klaus, The Fourth Industrial Revolution: what it means, how to respond, 2016.

[16] Toledo,T., Musicant,O., and Lotan,T., “In-vehicle data recorders for monitoring and feedback on drivers’ behavior”, *Transportation Research Part C: Emerging Technologies*, Vol.16, No.3, pp.320-331, 2008.

[17] Wouters, P.I.J. and Bos, J.M.J., “Traffic accident reduction by monitoring driver behaviour with in-car data recorders”, *Accident Analysis & Prevention*, Vol.32, No.5, pp.643-650, 2000.

[18] Wouters, P.I.J. and Bos, J.M.J., The impact of driver monitoring with vehicle data recorders on accident occurrence : methodology and results of a field trial in Belgium and The Netherlands, SWOV Institute for Road Safety Research (Leidschendam, Netherland), 1997.

[19] Y.Zhong, J.Han, T.Zhang, Z.Li, J.Fang, and G.Chen, “Towards parallel spatial query processing for big spatial data”, Parallel and Distributed Processing Symposium Workshops & PhD Forum (IPDPSW), pp.2085-2094, 2012.

[20] Zeng, G. “Application of Big Data in Intelligent Traffic System”, *IOSR Journal of Computer Engineering*, Vol.17, No.1, pp.01-04, 2015.

[21] http://www.bigdata.go.kr/intro.html.

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

[23] http://hbase.apache.org/.

[24] http://spark.apache.org/.

[25] http://zeppelin.apache.org/.

[26] https://nifi.apache.org/.

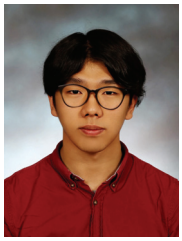
[27] https://www.intel.com/content/dam/www/public/us/en/documents/case-studies/big-data-xeon-e5-trustway-case-study.pdf.

[28] https://www.scylladb.com/2019/05/23/workload-prioritization-running-oltp-and-olap-traffic-on-the-same-superhighway/.

사 사

이 논문은 2019년도 정부(경찰청)의 재원으로 도로교통공단의 지원을 받아 수행하고, 한국교통안전공단 운수안전 컨설팅 사업의 지원을 받아 수행한 연구임(과제번호: 1325163906, 과제명: 자율주행을 위한 AI 기반 신호제어 시스템 개발).

저 자 소 개

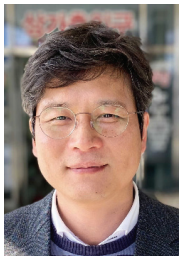


김 기 수(Ki-su Kim)

- 2017년 2월 : 서울시립대학교 공간정보공학과 (공학사)
- 2018년 8월 : 서울시립대학교 공간정보공학과 (공학석사)
- 2019년 12월~현재 : 디토닉 주식회사 컨설팅사업팀
- 관심분야 : 빅데이터 활용, 데이터 마이닝, 데이터 시각화, 데이터 분석

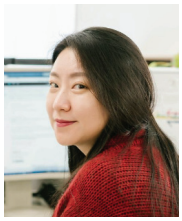
이 재 진(Jae-Jin Yi)

- 2009년 2월 : 아주대학교 국어국문과 (문학사)
- 2018년 ~현재 : 디토닉 주식회사
- 관심분야 : 시공간빅데이터, 빅데이터 솔루션 개발, 자율주행



김 홍 회(Hong-Hoi Kim)

- 1999년 2월 : 군산대학교 전자계산학과 (공학사)
- 2002년 8월 : 충남대학교 컴퓨터공학과 (공학석사)
- 2014년 7월~현재 : 디토닉 주식회사 컨설팅사업팀수석연구원
- 관심분야 : 교통빅데이터, 스마트시티, 자율주행자동차, MaaS



장 유 림(Yo-lim Jang)

- 2000년 2월 : 계명대학교 교통공학과 (공학사)
- 2003년 2월 : 서울대학교 지구 환경시스템공학부 (공학석사)
- 2013년 8월 : 서울대학교 지구 환경시스템공학부 (공학박사)
- 2016년 ~현재 : 한국교통안전공단 교통빅데이터센터, 책임연구원
- 관심분야 : 교통빅데이터

함 유 근(Yu-Kun Hahm)

- 1984년 2월 : 고려대학교 정경대학 통계학과
- 1994년 : Boston University (경영학 박사)
- 1998년 ~현재 : 건국대학교 경영대학 경영학과 교수
- 관심분야 : 빅데이터 활용 전략, 빅데이터 비즈니스 모델