JKIICE Journal of the Korea Institute of Information and Communication Engineering

한국정보통신학회논문지(J. Korea Inst. Inf. Commun. Eng.) Vol. 19, No. 3 : 607~612 Mar. 2015

Hadoop을 이용한 스마트 자동차 서비스용 빅 데이터 솔루션 개발

라이오넬[1] · 장종욱[2*]

Addressing Big Data solution enabled Connected Vehicle services using Hadoop

Lionel Nkenyereye[1] · Jong-Wook Jang[2*]

[1]Department of Computer Engineering, Dong-Eui University, Busan 614-714, Korea
[2*]Department of Computer Engineering, Dong-Eui University, Busan 614-714, Korea

요 약

자동차 진단 데이터의 양이 증가함에 따라 자동차 에코시스템의 액터는 스마트 자동차에서 수집된 데이터에 따라 새로운 서비스를 시뮬레이션 하거나 설계하기 위하여 실시간으로 분석을 해야 하는 어려움에 직면하게 된다. 본 논문에서는 자동차에서 생성된 막대한 양의 자동차 내장 진단 데이터를 처리하고 분석하는데 필수적이고 심오한 해석학을 제시하는 빅 데이터 솔루션에 관한 연구를 하였다. Hadoop 및 그 에코시스템은 자동차 소유자에 대한 새로운 서비스 제공을 위해 자동차 에코시스템의 액터에 의해 사용될 수 있는 막대한 데이터 및 전달된 유용한 결과를 처리하기 위해 개발된 것이다. 지능형 교통시스템이 안전성 보장, 속도로 인한 사고로 입는 상해 및 충돌의 비율 감소 등에 관여함에 따라, 자동차 진단 데이터 기반의 빅 데이터 솔루션 개발을 통해 향후 실시간 결과 감시, 여러 스마트 자동차에서의 데이터 수집, 수집된 데이터에 대한 신뢰성 있는 처리 및 용이한 저장을 실현화하게 된다.

ABSTRACT

As the amount of vehicle's diagnostics data increases, the actors in automotive ecosystem will encounter difficulties to perform a real time analysis in order to simulate or to design new services according to the data gathered from the connected cars. In this paper, we have conducted a study of a Big Data solution that expresses the essential deep analytics to process and analyze vast quantities of vehicles on board diagnostics data generated by cars. Hadoop and its ecosystems have been deployed to process a large data and delivered useful outcomes that may be used by actors in automotive ecosystem to deliver new services to car owners. As the Intelligent transport system is involved to guarantee safety, reduce rate of crash and injured in the accident due to speed, addressing big data solution based on vehicle diagnostics data is upcoming to monitor real time outcome from it and making collection of data from several connected cars, facilitating reliable processing and easier storage of data collected.

키워드 : 커넥티드 카, 하둡, 하이브 오픈소스, 빅데이터, 진단서비스

Key word : Connected car, Hadoop project, HIVE open source, Big data Problem, on-board diagnostics, MapReduce

접수일자 : 2014. 12. 11 심사완료일자 : 2015. 01. 15 게재확정일자 : 2015. 01. 30

* **Corresponding Author** Jong-Wook Jang(E-mail: jwjang@deu.ac.kr, Tel:+82-51-890-1709)
Department of Computer Engineering, Dong-Eui University, Busan 614-714, Korea

Open Access **http://dx.doi.org/10.6109/jkiice.2015.19.3.607** print ISSN: 2234-4772 online ISSN: 2288-4165

ⓒThis is an Open Access article distributed under the terms of the Creative Commons Attribution Non-Commercial License(http://creativecommons.org/li-censes/by-nc/3.0/) which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.
Copyright Ⓒ The Korea Institute of Information and Communication Engineering.

Ⅰ. INTRODUCTION

The information technology authorizes information sharing from vehicles, roads infrastructure’s sensors, vehicle&#8211;to vehicle, vehicle-to infrastructure, road condition [1]. Based on data received from ITS (Intelligent Transportation System), the third party interested in automobile ecosystem such as car manufacturers, repair shops, road and transportations authorities will continuously support a multitude of applications as for instance, monitoring performance of vehicles sold in the market by leveraging reporting of DTC(Diagnostic Troubles codes), status diagnosis information, emergency services management to locate and help victims of accidents or injured persons, traffic transportation authorities to increase safety, accident prevention, car repairs to analyze vehicle diagnosis in real time in case the vehicle breaks down.

However, data from connected cars will serve as a source to the actors in automotive ecosystem to afford value added services to the car owners. It seems that as long as data from connected vehicles increases, a big solution is expected to be implemented by car manufacturers or transportation authorities in order to process it and make available a reliable database of outcomes to all interested in connected car field. Apache Hadoop is nowadays suitable to process a huge amount of data. In this paper, we present a study of big data solution using Hadoop to process connected vehicle’s diagnostics data and the results from processing will allow connected vehicle services.

Ⅱ. RELATED STUDIES

2.1. Internet of vehicle and geographical navigation application program from Google

The evolution of IoT(Internet of Things) has brought one line many little nodes plainly exists today [2]. Google has considered the vehicle as one of the big node comprise by a lot of little node. Google 's Waze(the world’s largest community-based traffic and navigation application) division gathers traffic information from all enabled Waze geographical application based on smartphone out and distribute that information to all car's owner who has opted to use Google’s Waze.

Like other GPS(Global Positioning System) software it considers from users' driving times to provide routing and real-time traffic updates. People can report accidents, traffic jams, speed and police traps, and from the online map editor, they can update roads, landmarks, house numbers, etc. [3].

The Google’s Waze GPS application involves only car owners or drivers and requires to upload their current reports accidents in order to allow them to make their optimal navigation routing decisions. This appear to be a weakness of the connected car vehicle platform that enables sharing vehicle sensors data and allows others services providers interest to automotive industry to design and propose their respectively services to the car owners. The proposed solution in this paper highlights acquistion of vehicles on board diagnostics data, real time processing using Apache Hadoop. The outcomes from processing are stored back to the relational database or generated in CVS(Comma Separated Values) format, so actors in automotive ecosystem can access them via internet and provide new services to the car owners.

2.2. Apache Hadoop platform

Hadoop is an open source framework for writing and running distributed applications that process large amounts of data[4]. Data is broken up and distributed across the cluster and as much as possible, computation on a piece of data taken place on the same machine where the piece of data resides.

The Hadoop projects that are used in this paper to keeping processing vehicle sensor data are: HDFS (Hadoop Distributed File System) a distributed file system that runs on large clusters of commodity clusters, Apache Hive a distributed data warehouse that manages data stored in and make possible for analysts with strong SQL(Structured Query Language) skills.

2.3. Big data context

The big data refers to a collection of data sets that relational databases have are not able to store for example a few terabyte of unstructured data from a variety of sources can considered as big data [4].

Effectively analysis of data is considered as a new way to increase productivity and also accelerate new services which may be proposed to car owners.The evolution of IoT(Internet of Things) has brought one line many little nodes.

Ⅲ. SYSTEM DESIGN

3.1. Design of the solution to handle vehicle diagnostic data

The solution proposed is related to use Hadoop framework. After vehicle diagnostic software based on android performs an upload of vehicle diagnostics data, to the data centers, Hadoop framework will process them and the outcomes will be stored on the web server on which third party can access them. Fig. 1 shows the design of solution proposed based on Hadoop.

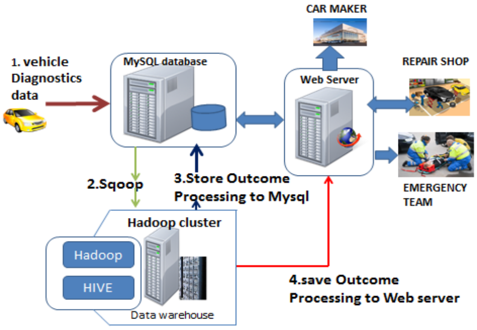


그림 1. 하둡 기반으로 구성한 솔루션의 개요
Fig. 1 Overview of our solution based on Hadoop

3.2. Acquisition of vehicle diagnostics data

The acquisition of vehicle diagnostics data describes in this paper consists of OBD(On-Board Diagnostics) scan tool, an android smartphone and a database (MySQL database) known to us as a RDBMS(Relational Database Management System). OBD scan tool interacts with the ECU(Engine Control Unit), while the android device is used as the client to handle data acquisition and data transmission, the database stores vehicles diagnostics data.

The HTTP(Hypertext Transfer Protocol) in this paper is used to complete the communication between android and web server. The auto scanner tool used to connect with OBD interface starts reading OBD-PIDs(On-board Diagnostics Parameters IDs) codes[5]. The profile of the driver as for instance car owner name, car owner password, and valid vehicle identification based on Bluetooth MAC(Machine Access Address) of the OBD scan tool are authenticated on the database, and then sent back to the car owner. The failure of authentication leads to a new car owner registration. Fig. 2 below summarizes the design of the acquisition of vehicle diagnostics data to the remote data centers.

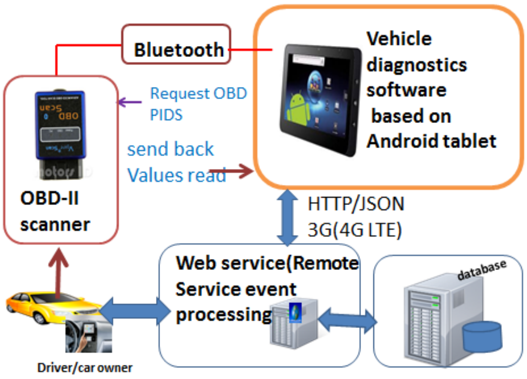


그림 2. 원격 데이터 센터에 차량 진단 데이터 수집 과정
Fig. 2 Overview of the acquisition vehicle diagnostics data to a remote data center

3.3. Processing of vehicle diagnostics data using Hadoop framework

The processing of the data acquired on the remote database is divided in four phases: import data from MySQL to Hadoop clusters, loading data from HDFS to Apache Hive, analysis through Hadoop MapReduce framework, upload outcome files in CVS format from HDFS to the web server.

1) import data from MySQL to Hadoop data file system(HDFS)

In this phase, the process consists of importing data from MySQL into Sqoop. Sqoop is an open-source tool that allows users to extract data from a relational database into Hadoop for further processing[6].

2) loading data from HDFS to Hive

After Apache Sqoop keeping parallelizing import across multiple mappers, the next step consists of loading data into a Hive table using Apache Sqoop. The import data into hive relies on sake of efficiency that has a post processing step where Hive table is created and loaded. When the data is loaded into HIVE from HDFS directory, Hive moves the sqoop replication table which is viewed as a directory into its warehouse rather than copying data.

3) analysis using Map Reduce framework

Hadoop acts as a distributed computing system that holds a distributed file system across Hadoop cluster. It relies to a distributed computational, MapReduce, which coordinates and runs jobs in the cluster to operate on a part of the overall processing task in parallel. Some of this analysis needs to be performed by using customs map and reduce scripts. In conjunction with HiveQL. (HiveQL is Hive's query language, a dialect of SQL) It is heavily influenced by MySQL[6], the developers can write SQL scripts to perform historical analysis over data loading in Apache Hive warehouse. SQL is great for many analyses, and it has the huge advantage of being very well known in the industry, so Hive is well placed to integrate with business intelligence tools (ODBC for example).

4) Save the outcome to the web server

In this paper, when the final result from processing is available, Apache Hive is able to copy the outcome to the web server, thereafter Sqoop can exports it back to the MySQL database.

Ⅳ. IMPLEMENTATION AND ITS RESULTS

The purpose of the big data solution is to process vehicle diagnostics data, save the outcomes into a MySQL database and on the web server on which data is remained available for the third party (actors in automotive ecosystem) for their further needs. Therefore, we set up a plan of required useful outcome information. For each useful information, we develop and execute a MapReduce jobs on single node. Fig. 3 describes in summary the result expected for each job according to the vehicle diagnostics data and on which event the outcomes may be used to fulfill car owners' needs or services.

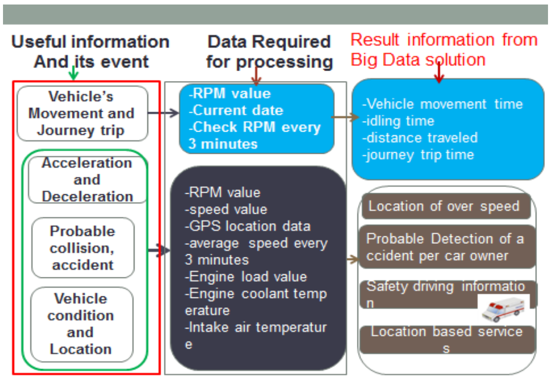
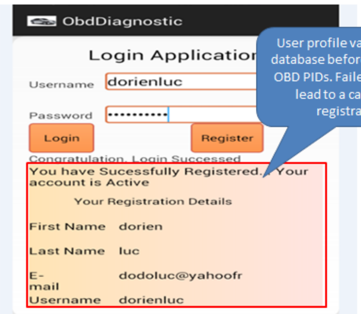

Useful information And its event
Data Required for processing
Result information from Big Data solution
Vehicle's Movement and Journey trip
Acceleration and Deceleration
Probable collision, accident
Vehicle condition and Location
-RPM value
-Current date
-Check RPM every 3 minutes
-Vehicle movement time
-idling time
-distance traveled
-journey trip time
-RPM value
-speed value
-GPS location data
-average speed every 3 minutes
-Engine load value
-Engine coolant temperature
-Intake air temperature
Location of over speed
Probable Detection of a ccident per car owner
Safety driving information
Location based services

그림 3. 각 맵리듀스에서 예상되는 처리 과정
Fig. 3 Summary of processing vehicle diagnostics data using Hadoop

4.1. Data acquisition and storage

In this paper, we develop a Remote Vehicle Diagnostics application based on android. It will be used by the vehicle owner via his mobile device based android. The car owner starts by connecting the Bluetooth OBD scan tool adapter into OBD car's connector. Once the connection with the android based smartphone is established, he starts request data and the action is performed in background services. Fig. 4 and Fig. 5 show login before requesting engine performance and values uploaded into database.



ObdDiagnostic

Login Application

Username dorienluc

Password ••••••••••

Login Register

Congratulation, Login Successed

You have Sucessfully Registered..Your account is Active

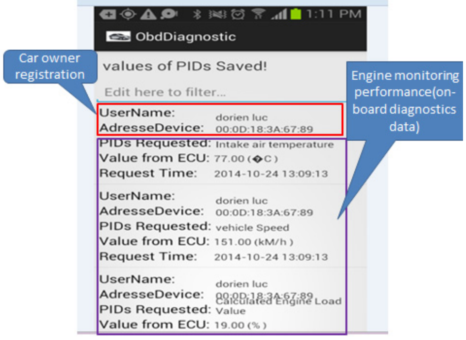
Your Registration Details

First Name	dorien
Last Name	luc
E-mail	dodoluc@yahoofr
Username	dorienluc

User profile validation on database before requesting OBD PIDs. Failed of profile lead to a car owner registration

그림 4. OBD-PID를 요청하기 전 클라우드에서 운전자 프로필 검증화면

Fig. 4 Validation car user profile on cloud before requesting OBD-PIDs



1:11 PM

ObdDiagnostic

values of PIDs Saved!

Edit here to filter...

UserName: dorien luc
AdresseDevice: 00:0D:18:3A:67:89
PIDs Requested: Intake air temperature
Value from ECU: 77.00 (�C)
Request Time: 2014-10-24 13:09:13

UserName: dorien luc
AdresseDevice: 00:0D:18:3A:67:89
PIDs Requested: vehicle Speed
Value from ECU: 151.00 (kM/h)
Request Time: 2014-10-24 13:09:13

UserName: dorien luc
AdresseDevice: 00:0D:18:3A:67:89
PIDs Requested: Calculated Engine Load Value
Value from ECU: 19.00 (%)

Car owner registration

Engine monitoring performance(on-board diagnostics data)

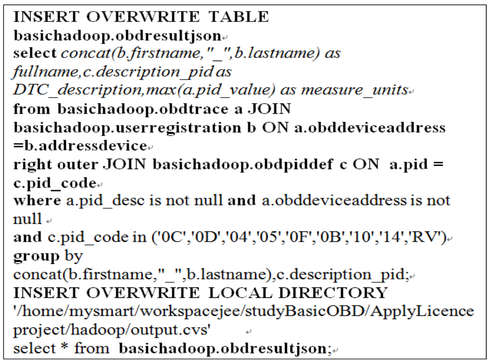
그림 5. 운전자가 클라우드에 저장된 on-board 진단 데이터 요청화면

Fig. 5 Request of on-board diagnostics data saved on the cloud by the car owner

4.2. Processing using Hadoop platform

Apache Hadoop open source and its projects are used to process the on-board diagnostics data and make available useful information to the third party as described in (see the Fig. 3). When on-board diagnostics data are uploaded to the database, Apache Sqoop performs a replication import of data required to run Map Reduce jobs. Before the import can be initiated, Sqoop uses JDBC(Java Database Connectivity)to examine the table to import [6]. Apache Hive open source has an important role especially for storing vehicle diagnostics data unto a relational database. Sqoop generates a Hive table based on it originally definition known as metadata and at the same time stores data on HDFS. Data are imported into Apache Hive data warehouse and stored on HDFS as well.

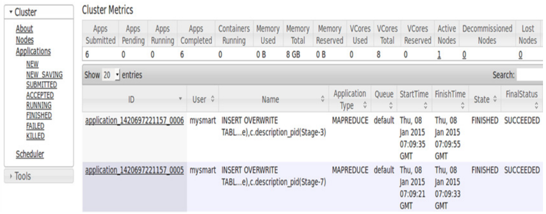
One of the most key of Map Reduce jobs implementation is to take the vehicle on-board diagnostics data replicated from MySQL database and stored into Hive warehouse; process them according to the useful information we design and then store it to the MySQL database or generate it into CVS, JSON(JavaScript Object Notation) formats that can be readable by users . HiveQL is used to execute Map Reduce jobs. Fig. 6 shows our UDF(User-Defined Functions) pseudocode using HiveQL query. In this paper, we consider that the value of the RPM(Rate of Revolution of a Motor) for example helps to compute vehicle's movement by taking in consideration current data and value of RPM after every 3 minutes. Fig. 7 and Fig. 8 show how Hadoop converts HiveQL queries into a set of Map Reduce jobs.



```
INSERT OVERWRITE TABLE
basichadoop.obdresultjson
select concat(b.firstname,"_",b.lastname) as
fullname,c.description_pid as
DTC_description,max(a.pid_value) as measure_units
from basichadoop.obdtrace a JOIN
basichadoop.userregistration b ON a.obddeviceaddress
=b.addressdevice
right outer JOIN basichadoop.obdpiddef c ON  a.pid =
c.pid_code
where a.pid_desc is not null and a.obddeviceaddress is not
null
and c.pid_code in ('0C','0D','04','05','0F','0B','10','14','RV')
group by
concat(b.firstname,"_",b.lastname),c.description_pid;
INSERT OVERWRITE LOCAL DIRECTORY
'/home/mysmart/workspacejee/studyBasicOBD/ApplyLicence
project/hadoop/output.cvs'
select * from  basichadoop.obdresultjson;
```

그림 6. HiveQL 쿼리 사용한 맵리듀스 기능 정의 구조

Fig. 6 Map Reduce function definition structure using HiveQL query



Cluster Metrics

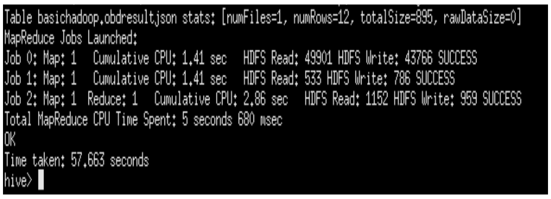
Apps Submitted	Apps Pending	Apps Running	Apps Completed	Containers Running	Memory Used	Memory Total	Memory Reserved	VCores Used	VCores Total	VCores Reserved	Active Nodes	Decommissioned Nodes	Lost Nodes
6	0	0	6	0	0 B	8 GB	0 B	0	8	0	1	0	0

Show 20 entries Search:

ID	User	Name	Application Type	Queue	StartTime	FinishTime	State	FinalStatus
application_1420697221157_0006	mysmart	INSERT OVERWRITE TABL...e),c.description_pid(Stage-3)	MAPREDUCE	default	Thu, 08 Jan 2015 07:09:35 GMT	Thu, 08 Jan 2015 07:09:55 GMT	FINISHED	SUCCEEDED
application_1420697221157_0005	mysmart	INSERT OVERWRITE TABL...e),c.description_pid(Stage-7)	MAPREDUCE	default	Thu, 08 Jan 2015 07:09:21 GMT	Thu, 08 Jan 2015 07:09:33 GMT	FINISHED	SUCCEEDED

그림 7. 맵리듀스 작업 집합으로 하둡 converted에 HiveQL 쿼리 실행

Fig. 7 HiveQL query runned on Hadoop converted into a set of Map Reduce jobs



```
Table basichadoop.obdresultjson stats: [numFiles=1, numRows=12, totalSize=895, rawDataSize=0]
MapReduce Jobs Launched:
Job 0: Map: 1   Cumulative CPU: 1.41 sec   HDFS Read: 49901 HDFS Write: 43766 SUCCESS
Job 1: Map: 1   Cumulative CPU: 1.41 sec   HDFS Read: 533 HDFS Write: 786 SUCCESS
Job 2: Map: 1  Reduce: 1   Cumulative CPU: 2.86 sec   HDFS Read: 1152 HDFS Write: 959 SUCCESS
Total MapReduce CPU Time Spent: 5 seconds 680 msec
OK
Time taken: 57.663 seconds
hive> 
```

그림 8. 하나의 클러스터 노드에서 실행된 맵리듀스 기능
Fig. 8 Map Reduce function in execution on single cluster node

Ⅴ. CONCLUSION AND FUTURE WORK

With the development of Hadoop platform project, now is possible to build big data solution using open source projects integrated with Hadoop. In this paper, a study of big data solution for processing data from vehicles using Hadoop and making final results available, allows accessing useful information via web services to the third party such as car manufacturer, transportation and road operators, car dealers, police, emergency services has been conducted on a single node cluster. The outcome obtained from various Map Reduce functions managed after executing HiveQL query indicate favorable results in term of time taken.

Future work will center on implementing and evaluating performance on cloud platforms for instance EC2(Amazon Elastic compute Cloud).

ACKNOWLEDGMENT

This work was supported by Dong-eui University Grant.(No.2014AA324), and Nurimaru R&BD(No. 2014XC005)

REFERENCES

[1] Jukka Ahola. "Vehicle services opportunities benefit from the cloud", *Applying cloud technologies for business Magazine*, page78-79.

[2] Google's Driverless Car, The Internet of Things, And Georges Orwell. Available : http://www.forbes.com/sites/rogerkay/2014/09/08/googles-driverless-car-the-internet-of-things-and-george-orwell/

[3] Waze Social GPS, Maps & Traffic. Available: https://itunes.apple.com/us/app/waze-social-gps-maps-traffic/id323229106?mt=8

[4] Aditya B.Patel,Manashvi Birla,Ushma Nair,"Addressing Big Data Problem using Hadoop and Map Reduce",*2012 Nirma University International Conference On Engineering, nuicone-2012*,06-08 December, 2012.

[5] Wikipedia, OBD-II PIDs, Available: http://en.wikipedia.org/wiki/OBD-II_PIDs.

[6] Alex Holmes, "Hadoop in Practice, Including 85 techniques", chapter 4, "Applying MapReduce Pattern to Big data", page 139-173, ISBN:9781617290237.

장종욱(Jong-wook Jang)

1995년 2월 부산대학교 컴퓨터공학과 박사
1987년 ~ 1995년 ETRI
2000년 2월 UMKC Post-Doc.
1995년 ~ 현재 동의대학교 컴퓨터공학과 교수
※관심분야 : 유무선통신시스템, 자동차네트워크

라이오넬(Lionel Nkenyereye)

동의대학교 컴퓨터공학과 석사과정
※관심분야 : 유무선통신네트워크, 빅데이터, 안드로이드 시스템