원격 자동차 고장 진단 시스템 개발에 대한 연구

라이오넬* · 장종욱*

*동의대학교

A study on Development of Remote Vehicle Fault Diagnostic System

Lionel Nkenyereye* · Jong-Wook Jang*

*Dong-Eui University

E-mail : lionelnk82@gmail.com*, jwjang@deu.ac.kr*

요 약

일반적으로 자동차드라이버의 스마트폰을 통한 데이터전송은 자동차운전자의 핸드폰은 데이터를 실시간으로 원격데이터 센터에 전송하는 경우에 용량 의존적인 순위를 가지고 있다. 생성되는 진단 보드 데이터들은 드라이버의 폰에서의 모바일 진단 어플리케이션에 임시적으로 저장하고, 인터넷에 연결 되었을 때 데이터 센터에 전송한다. 클라우드에서 실행에 방해하는 다른 태스크들이 없는 원격 자동차 어플리케이션 사용방법을 위한 node.js는 모바일 네트워크을 통한 클라우드에서 데이터 저장 업무를 다루기 위하여 적합하다. 우리는 외부 어플리케이션으로부터 driver inputs and delivers output을 패스하는 원격 유저와 운용하는 스마트폰 어플리케이션에서 자동차와의 어플리케이션 interface 방법을 사용하는 실시간 분석 안드로이드 어플리케이션 반응을 시뮬레이션 통해 제안된 아키텍쳐의 유효성을 입증한다.

이 논문에서, 우리는 이벤트 루프 접근을 기반으로 하는 이것은 웹서버 구조를 특징으로 하는 원격 자동차 결함 진단 시스템 연구를 제안한다.

ABSTRACT

Data transmission via the car driver’s tethered smart phone may have a volume-dependent billing in case car driver’ phone transmits data in real-time to the remote data center. The on-board diagnosis data generated are temporary stored locally to mobile remote diagnosis application on the car driver’s phone, and then transmit to the data center later when car driver connects to the Internet. To increase the easiest of using the remote vehicle application without blocking other tasks to be executing on the cloud, node.js stands as a suitable candidate for handling tasks of data storage on the cloud via mobile network. We demonstrate the effectiveness of the proposed architecture by simulating a preliminary case study of an android application responsible of real time analysis by using a vehicle-to- smart phones applications interface approach that considers the smart phones to act as a remote user which passes driver inputs and delivers output from external applications. In this paper, we propose a study on development of Remote Vehicle fault diagnostic system features web server architecture based event loop approach using node.js platform, and wireless communication to handle vehicle diagnostics data to a data center.

Keyword

event-driven approach, remote vehicle diagnostics system, node.js, web architecture, Connected vehicle, On-Board Diagnostics Parameters-IDs(OBD-PIDs)

Ⅰ. Introduction

Fault detection aptitude is essential for the survivability of a transportation system and its users.

Data transmission cost will be a relevant factor for the car owner in case the car driver‘s phone is always connected to the car’s diagnostics and continually transmit them to the remote data center. Beside the cost communication, the volume of data to be stored at the remote is huge, so processing them and enables quickly decision making require a powerful distributed computational system such as Hadoop or storing data into NoSQL database. In addition, the design of remote mobile application should guarantee that the computation of tasks may not prevent car owner to use other application, so an event loop web server on the cloud such as node.js is suitable than a multi thread web server architecture.

In this paper, we propose a study on development of remote vehicle diagnostics service that will allow a reduce cost of data communication and enables an asynchronous transmission of the car’s diagnostics and current trouble code to the remote data center via node.js after a certain time of driving or when car driver uses others mobile applications in order to save communication cost.

Ⅱ. Communication Cost challenge and Data volume for Remote Vehicle Diagnosis Service

Data transmission of car’s diagnostics via car driver’s tethered smartphone always connected to the car’s diagnostics system located near the car driver seat and transmitting data to the remote data center increase cost in communication. The data transmission may not only increase cost communication but may also limit the use of other applications in case the architecture of the web server responsible of handling tasks from the mobile device to the database for instance allows execution of all the works for each request on individual threads, the car owner will interrupt the data transmission when he (she) likes use other applications installed on his(her) smart phone. In addition, the data volume will gradually increase day after day, and then challenge the processing tasks and decision making. On the cloud computing side, the relational database is not suitable because of its inheriting design of schema and storage constraints. To increase the easiest of using the remote vehicle application without blocking other tasks to be executing on the cloud, node.js stands as a suitable candidate for handling tasks of data storage on the cloud via mobile network after the application have temporary stored locally to the android mobile private database for instance SQLite database management. The connectivity inside the vehicle may be established with the network operator in three ways: embedded solution, Tethered solution and integrated solution [2].

Ⅲ. System Design for Remote Vehicle Fault Diagnosis

This study developed a remote diagnostic system based on OBD-II and remote datacenter where the JavaScript runtime environment running Google chrome’s V8 engine known as Node.js for the automotive engine. Figure 1 shows that this system is able to verify engine information and existence of malfunction there in by Bluetooth communication between the ECUs using OBD-II protocol. The engine information and diagnostics troubles codes are temporary stored on the proposed application, then transmits to the remote data center when the user connects to the internet or activates internet connection while is driving. The Node.js is responsible for handling data transmission to a NoSQL database.

3.1 Bluetooth OBD-II protocol structure for engine status information collection

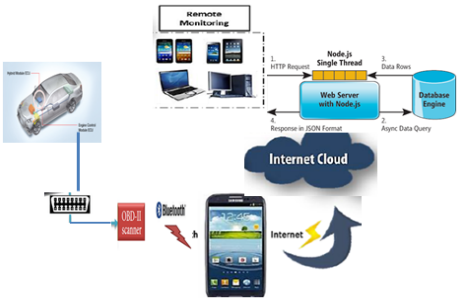

Fig. 1 System design of mobile vehicle diagnostics system with Node.js at web side

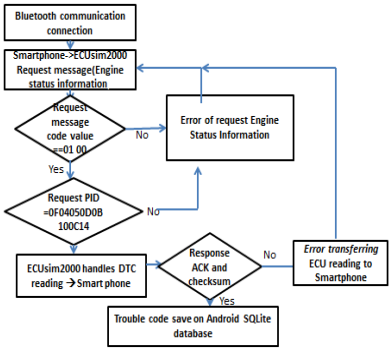

Fig. 2 Flowchart of engine status information collection algorithm

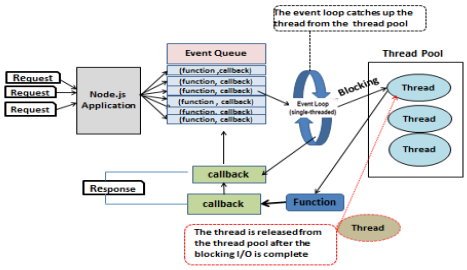

Fig. 3 Node.js server side for handling blocking I/O request using event Queue

The OBDII is a standard that diagnoses the information on main system of vehicles or on failure transmitted from sensors attached to a vehicle to ECU from a center console or external device by using the serial communication function [3]. Figure 2 is proposed OBD-II protocol for this study.

3.2 Web server based Node.js for handling car's diagnostics system

The event-driven based model is popular known to its capability to assign asynchronous and non-blocking call semantics [5].

The node.js stands for reliable web server architecture due to the asynchronous/non-blocking call semantics which features thread-per-connection model. The model is the mapping of a single thread to multiple connections. The thread then handles all occurring events from I/O operations of these connections and requests as shown on the Figure 3.

Ⅳ. Implementation of Remote Vehicle Diagnosis Fault and its results

Once the mobile remote mobile application discovers the remote cloud web server with Node.js, it starts request data and the action is performed in background services. Figure 4 shows a screen shot while the car owner monitors on-board diagnostics saved on the SQLite database management based on the android during his trip on his (her) mobile device before transferring them to the remote. Figure 5, shows establishment of communication by the node.js server side client between the mobile device and the web server based node.js host. Figure 6 shows car users receives fault trouble from the remote diagnostic data center to their smart phone in the notification area

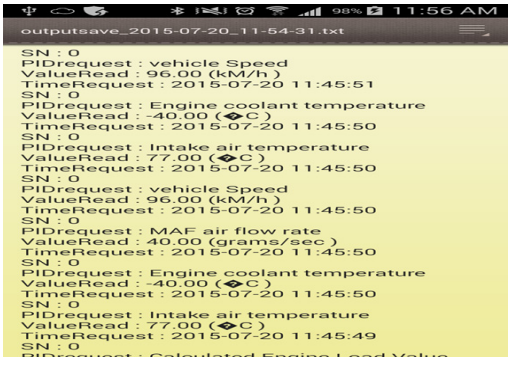

Fig. 4 File of Vehicle performance data saved on the SQLite database management on the car owner mobile device

Fig. 5 Discovering of Remote vehicle web server based Node.js for client establish communication

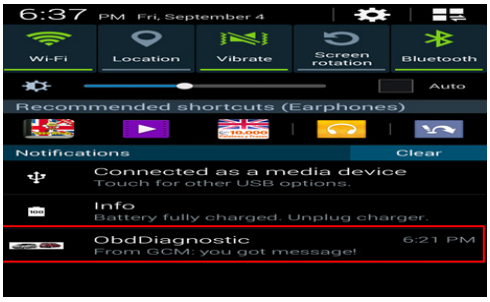

Fig. 6. Car User notification message of engine's fault .

Ⅴ. Conclusion

In this paper, we argue that concerns about uprightness of data from traffic and on-board diagnostics are a major step for vehicle owners, authorities and businesses looking to take up cloud computing, nomadic smartphones that enable telematics services and others value-added services. By using OBD-II protocol, a smart phone engine diagnostic system using Bluetooth communication was developed. In this study, instead of handling information that can be controlled only by car owners, it was made possible to select necessary information only and take control at first hand, then transferred them to the remote vehicle diagnosis data center at reduce cost when driver rest use his smartphone for other purpose while the node.js handling , storing generated car's status information to a MongoDB without waiting the remote vehicle diagnostics mobile Apps finishing the upload process submitted to node.js. A remote vehicle diagnostics software as a service attests the concept of vehicle to cloud capable of collecting diagnostics data. Therefore, with this system it was made possible that information of car's diagnostic system condition may be identified in real time and that if engine has malfunction, by notifying diagnostic trouble codes and information, the user and car manufacturer's experts may promptly respond to such malfunction.. Our next purpose is to implement a fully prototype to evaluate others value-added services.

Acknowledgment

이 논문은 2015년도 Brain Busan 21사업과 2015년도 누리마루사업에 의하여 지원되었음

References

[1] Minyoung, K., & Jang-Wook, J. (2012). Design of Korea smart car driving information checking system. International Journal of Advanced Smart Convergence.1(1)PP:38-42

[2] GsmamAutomotive, (2013). Connecting Cars:Bring your Own Device-Tethering Challenges. Report on Intelligent Trasporatation system Report, PP: 1-20

[3] D. J.Oliver, "Implementing the J1850protocol," http://smartdata.usbid.com/datasheets/usbid/2000/2000-q4/j1850 wp.pdf."

[4] Wikipedia CAN bus, http://en.wikipedia.org/wiki/CAN bus

[5] Benjamin, E. (2012). Concurrent Programming for Scalable web Architectures : Diploma Thesis Faculty of Engineering and Computer Science, Institute of Distributed System,PP:45-67.