Prototyping Commodity ICT for Mobility CPS

Yuki Iida[†], Yusuke Fujii[†], Takuya Azumi[†], Nobuhiko Nishio[†],
Shinpei Kato[*], Tsuyoshi Hamada[‡], Satoshi Kagami[§], and Kazuya Takeda[*]

[*] School of Information Science, Nagoya University
[†] College of Information Science and Engineering, Ritsumeikan University
[‡] Nagasaki Advanced Computing Center, Nagasaki University
[§] Digital Human Research Center, AIST

***Abstract*—The next-generation mobility infrastructure requires computer systems to be more and more intelligent and interactive with the physical world. A grand challenge to this cyber-physical systems (CPS) problem can be found in real-time computing of autonomic control and environmental perception. Given the volume of information in the physical world, the computational cost of autonomic control and environmental perception is likely very high, which may not be affordable for embedded mobile devices. In this paper, we explore a possibility of leveraging cloud computing technology to support such computationally expensive operations of intelligent mobility CPS applications. Specifically we quantify the overhead of offloading computations to the cloud using commodity information and communication technology (ICT) platforms taking an example of autonomous driving. This quantification of overhead plays a vital role in the design issue of emerging mobility CPS applications. Our experimental results demonstrate that the current communication standards including WiFi and LTE achieve sufficient throughput and latency even when used with mobile smartphones to transfer large images and high-rate control commands over the network.**

***Index Terms*—Cloud Computing; Smart Devices; CPS**

I. Introduction

Mobility is an essential piece of our life. In recent years, transportation systems and automobiles are becoming more and more intelligent underlying high-efficient mobility of the societal infrastructure. Such innovations in mobility will reduce car accidents, remove humanś stress, improve traffic throughput, and create new markets. To this end, the next-generation mobility technology is expected to achieve a tight coordination of computer systems and physical elements, often referred to as cyber-physical systems (CPS), facilitating a cyber-understanding of the physical world. A core challenge to this mobility CPS, however, is a power and space constraint of each mobility component. For example, moving vehicles cannot accommodate rich computer systems due to the limited battery power. Particularly the computational cost with respect to understanding the physical world is very high and the computing capability of current vehicular systems is woefully inadequate. In order to address this trade-off, mobility CPS must seek for a cooperative solution with advanced information and communication technology (ICT) such as cloud computing.

A good example of compute-intensive mobility CPS applications is an autonomous vehicle [1], [4], [6]. It must recognize roads, traffic signs, surrounding vehicles, and pedestrians in real-time. An autonomous vehicle in the current state of the art tends to use laser sensors and/or cameras for those environmental perception tasks. The laser sensors can detect object edges as a set of 3-D points by hardware, reducing the computational requirement imposed on the vehicular system software, but are generally very expensive way beyond consumer electronics prices. On the other hand, the cameras are less expensive in price but are available at the expense of computational cost, because image processing is highly compute-intensive. It would require a rich set of multicore CPUs and hardware accelerators such as GPUs to meet the desired frame rate. Unfortunately these devices may not be affordable for battery-operated vehicular systems due to power consumption issues. As mentioned earlier, therefore, we should seek for a possibility of leveraging cloud computing technology to offload compute-intensive tasks onto high-performance computing (HPC) servers over the network. A question raised herein is "what is the overhead of offloading computation and associated data to the cloud?" If this overhead is acceptable, commodity ICT platforms will be a strong basis for mobility CPS applications.

Contribution: We present a prototype implementation of commodity ICT platforms for mobility CPS applications to quantify the overhead of offloading computation and data to the cloud. Specifically we use a smartphone as an example of commodity ICT platforms to capture images and control an autonomous vehicle as an example of mobility CPS applications. The overhead of transferring images over the network governs the frame rate of image processing in the cloud while that of transferring control commands determines the minimum feedback-control period of autonomous driving. We demonstrate that the bottleneck of networked image processing can be found in the computation time itself rather than the network communication overhead. We also find that the average network throughput of commodity ICT is sufficient to execute autonomic control but the worst-case latency must be bounded to provide stability. This is a useful insight into a coordination of commodity ICT and mobility CPS.

Organization: The rest of this paper is organized as follows. Section II describes the basic concept behind this paper. Section III presents our prototyping of ICT platforms for an autonomous vehicle as an example of mobility CPS applications. Section IV provides the evaluation of overhead

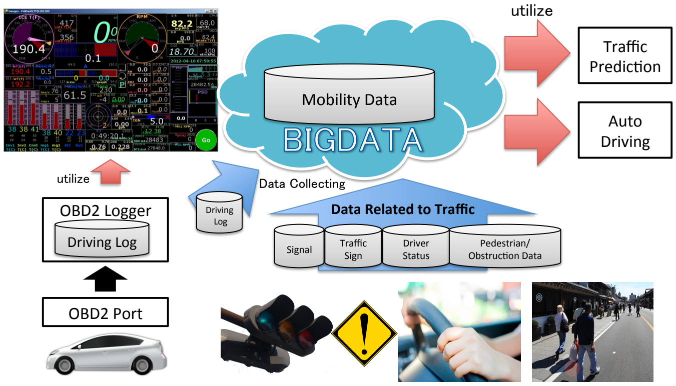

Fig. 1. Collecting automotive and environmental data.

imposed on data transfers over the network relevant to remote vehicle control and networked image processing. This paper concludes in Section V.

II. Basic Concept

In this paper, we advocate a concept of mobility CPS in the cloud. Many mobility CPS applications are battery-operated. They cannot accommodate large power consumers unless they equip special battery facilities. Integrating cloud technology allows computations and data to be offloaded over the network and we are freed from power consumption issues. For simplicity of description, we present our concept in the context of autonomous driving, but it is highly applicable for other mobility CPS applications.

Grand challenges of mobility CPS include a modeling of the physical world that underlies a real-time understanding of the physical world including enviromental perception and motion control. This modeling of the physical world requires accumulative collection of real-world data, often referred to as “Big Data”.

Fig. 1 illustrates an example of collecting automotive data through the on-board diagnosis (OBD) connectors as well as environmental data. We are building this system using commodity ICT products. The automotive data can be obtained through the CAN bus, while the environmental data can be captured using mobile devices such as smartphones. These data sets can actually be shared with a lot of mobility CPS agents. Such “Big Data” trends also encourage the concept of mobility CPS in the cloud. Since the data sets are stored in the cloud, each mobility CPS agent needs to access the network. For example, we can store a very large global trained data set [5] for anonymous agents to perform image recognition. The first step towards this approach is to understand the scale of latency and overhead imposed on cloud computing in real-time.

III. Prototyping

In this paper, we provide a prototyping of ICT platforms for mobility CPS applications, particularly taking an example of autonomous driving. We apply the cloud computing paradigm

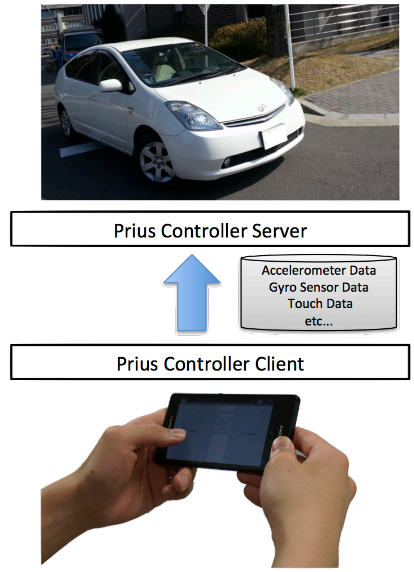

Fig. 2. A smartphone application for remote vehicle control.

for autonomous driving to mitigate high computational cost imposed on wimpy embedded vehicular systems. One of the major goals of this cloud-based autonomous driving system is to offload the computation requirement of autonomic control and environmental perception to a remote HPC server from a local vehicular system. Note that we focus on ICT platforms. Implementations of autonomic control and environmental perception for autonomous driving are outside the scope of this paper. Interested readers are encouraged to refer to our different contributions [2], [3].

In the rest of this section, we present two ICT platforms for the development of cloud-based autonomous driving. The first platform is a smartphone application that controls the vehicle from remote sources. The second platform is also a smartphone application that transfers captured images to the HPC server as fast as possible.

A. Remote Vehicle Control

We have a TOYOTA Prius that is modified to be able to overwrite the control of steering, accel, and break from the computer. This computer called “local master computer” is connected to an additional embedded board that can directly send signals to the inside wire system to control the vehicle. We omit a detailed description of this autonomous driving system, as the primary focus of this paper is the measurement of overhead and latency.

Fig. 2 illustrates the conceptual architecture of our experimental remote vehicle control system. We use a commodity smartphone to send control commands to the local master computer equipped within the vehicle through the network. The smartphone application determines the steering angle from the gyro sensor while the accel and the break strokes are controlled by the graphics user interface. Thus, we can intuitively use a smartphone as if it were a game controller of the vehicle remotely.

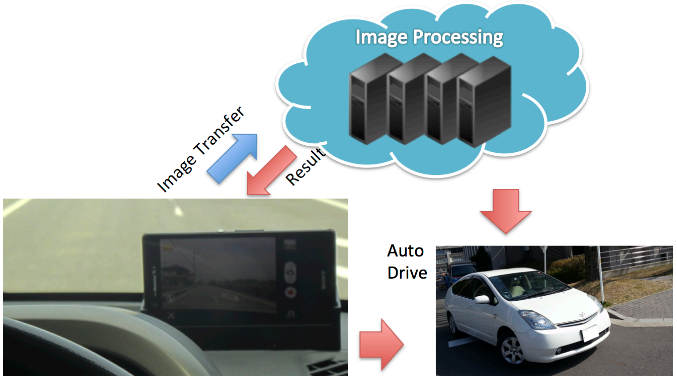

Fig. 3. Networked image processing.

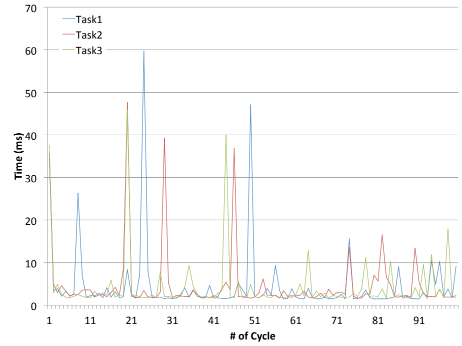

Fig. 5. The achieved period of *synchronous* data transfers for control commands using WiFi.

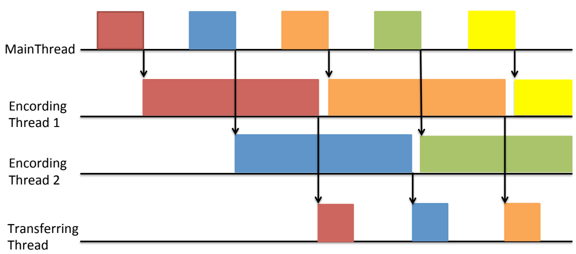

Fig. 4. Multithreading of image transfers.

B. *Networked Image Processing*

The computional cost of environmental perception in mobility CPS is highly expensive. The perception algorithm is often computationally complex and the volume of input data from laser sensors and cameras is not afforable for mobile embedded systems. Therefore it would be significant if we could offload these compute tasks to the cloud.

Fig. 3 illustrates the conceptual architecture of our experimental networked image processing system. We capture images in real-time from commodity smartphones attached in the vehicle and transfer them over the network to the HPC server where the actual image processing tasks are executing. The results of image processing such as the detected vehicles and pedestrians are fed back to the vehicular system. Though we have implemented many variants of image processing algorithms [2], they are not focused on in this paper. Instead we investigate if current commodity ICT platforms can transfer data over the network while meeting the desired throughput and latency.

Unlike control commands, the size of an image is large and it generates additional latency for the transfer. For example, each image transfer is composed of (i) capturing an image, (ii) encoding that image, and (iii) transferring the encoded image. Since these three stages can be pipelined, the total throughput would benefit from multithreading and a multicore processor. Furthermore, we have observed that the encoding stage is much longer than the capture and the transfer stages when we use a smartphone as a client due to its wimpy embedded processor performance. This means that the effective rate of image capture and transfer may be limitted to the execution time of encodling. In order to maximize the image transfer throughput on an embedded multicore processor, we increase the number of threads for encoding as shown in Fig. 4. In this example, one can see that the second (blue) transfer does not have to stall before the encoding stage because another thread is available for encoding whereas it would stall due to the preceding encoding process if there is only one thread available for encoding. Thus, we suggest that networked image processing using smartphones should make use of the multithreading capability and the parallelism of embedded multicore processors.

IV. Evaluation

We now evaluate the data transfer overhead in the context of cloud-based autonomous driving. We assume that autonomic control and environmental perception are provided in the remote server, but they are not within the scope of this paper. What we focus on in this experiment is the measurement of the data transfer overhead. Throughout the experiment, we assume that WiFi (IEEE802.11n 2.4GHz/5.0GHz) is provided with bandwidth of 300Mbps while LTE (au 2.1GHz LTE) achieves 75Mbps for downstream and 25Mbps for upstream.

A. *Control Command Transfer*

This experiment measures the transfer time taken to send control commands to the vehicle from a smartphone. The commands control the steering, accel, and brake of the vehicle. The steering angle is determined by the gyro sensor data while the accel and break strokes are manipulated by the graphics user interface of the Andrive application. We use the HTC J butterfly Snapdragon S4 Pro (APQ8064@1.5GHz, Quad Core) for a testing smartphone.

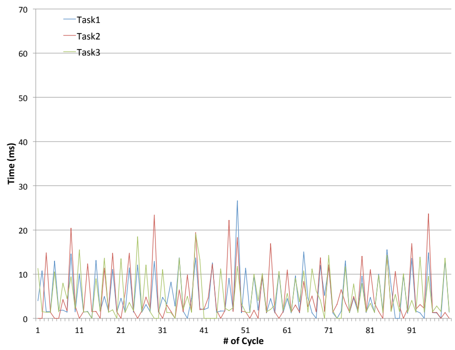

Fig. 6. The achieved period of *asynchronous* data transfers for control commands using WiFi.

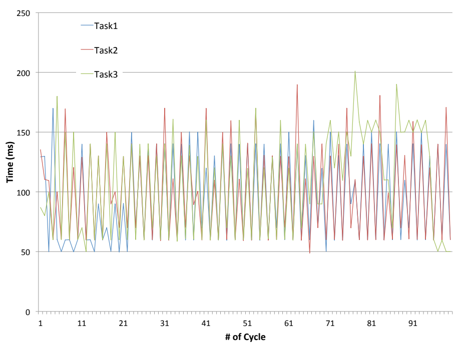

Fig. 7. The synchronous transfer time for control commands using LTE.

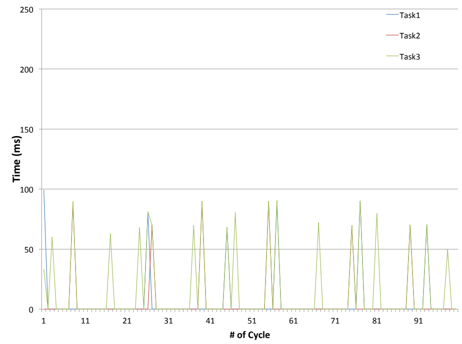

Fig. 8. The asynchronous transfer time for control commands using LTE.

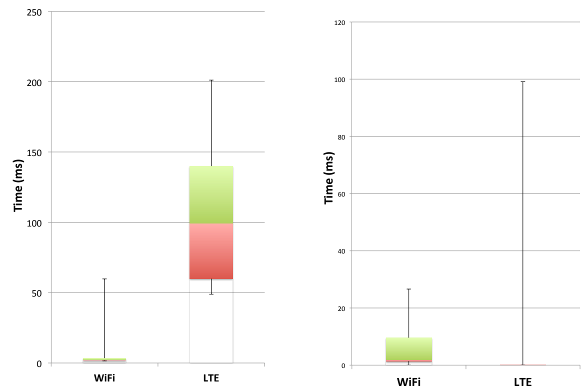

Fig. 9. Summarized box plotting of the transfer times.

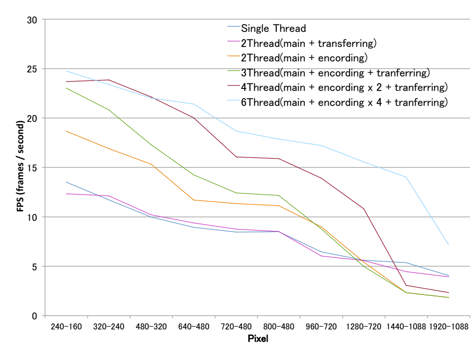

Fig. 10. The frame rate of networked image processing using WiFi.

Fig. 5 shows the period (interarrival time) of data transfers for vehicle control commands achieved using WiFi when the local master computer in the vehicle and a smartphone client are synchronized. The nodes are connected within a local area network, and the local master computer sends back an acknowledge message to the smartphone client every time the commands are received for synchronization. It meets a period of $5ms$ on average, which is acceptable for the feedback control rate of autonomous driving [3]. However there are unpredictable spikes that increase the period up to $60ms$ in the worst case. These unpredictable spikes are not acceptable under real-time constraints.

Fig. 6 shows

V. Conclusion

References

[1] E. Guizzo11. How Google's Self-Driving Car Works. IEEE Spectrum, 2011.

[2] M. Hirabayashi, S. Kato, M. Edahiro, K. Takeda, T. Kawano, and S. Mita. GPU Implementations of Object Detection using HOG Features and Deformable Models. In *Proc. of the IEEE International Conference on Cyber-Physical Systems, Networks, and Applications*, 2013 (under review).

[3] S. Kagami, S. Thompson, I. Samejima, N. Hatao, Y. Hihei, T. Egawa, T. Hamada, S. Kato, M. Kabasawa, H. Takemura, and H. Mizoguchi. Autonomous Vehicle Navigation by Building 3D Map and by Detecting Humand Trajectory using LIDAR. In *Proc. of the IEEE International Conference on Cyber-Physical Systems, Networks, and Applications*, 2013 (under review).

[4] J. Levinson, J. Askeland, J. Becker, J. Dolson, D. Held, S. Kammel, J.Z. Kolter, D. Langer, O. Pink, V. Pratt, M. Sokolsky, G. Stanek, D. Stavens, A. Teichman, M. Werling, and S. Thrun. Towards Fully Autonomous Driving: Systems and Algorithms. In *Proc. of the IEEE Intelligent Vehicles Symposium*, pages 163–168, 2011.

[5] H. Niknejad, T. Kawano, M. Simizu, and S. Mita. Vehicle detection using discriminatively trained part templates with variable size. In *Proc. of the IEEE Intelligent Vehicles Symposium*, pages 766–771, 2000.

[6] C. Urmson, J. Anhalt, H. Bae, D. Bagnell, C. Baker, R. Bittner, T. Brown, M. Clark, M. Darms, D. Demitrish, J. Dolan, D. Duggins, D. Ferguson,

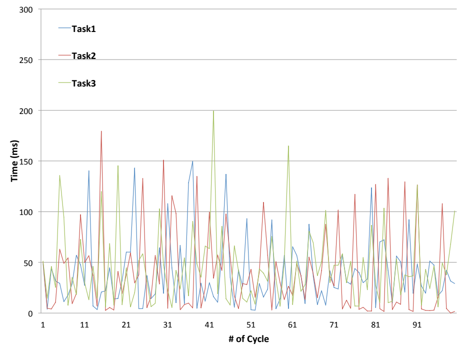

Fig. 11. The arrival time of networked image processing using WiFi.

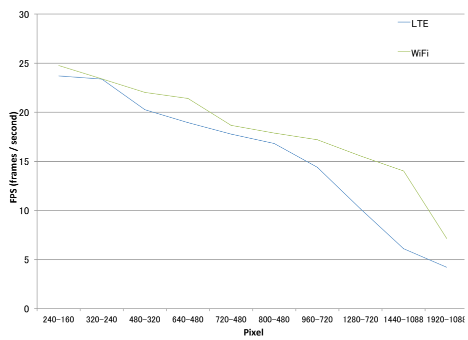

Fig. 12. The frame rate of networked image processing using LTE.

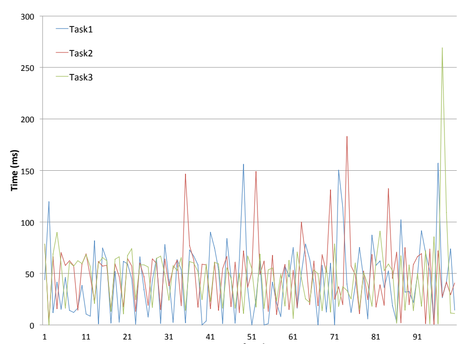

Fig. 13. The arrival time of networked image processing using LTE.

T. Galatali, C. Geyer, M. Gittleman, S. Harbaugh, M. Hebert, T. Howard, S. Kolski, M. Likhachev, B. Litkouhi, A. Kelly, M. McNaughton, N. Miller, J. Nickolaou, K. Peterson, B. Pilnick, R. Rajkumar, P. Rybski, V. Sadekar, B. Salesky, Y-W. Seo, S. Singh, J. Snider, J. Struble, A. Stentz, M. Taylor, W. Whittaker, Z. Wolkowicki, W. Zhang, and J. Ziglar. Autonomous Driving in Urban Environments: Boss and the Urban Challenge. *Journal of Field Robotics*, 25(8):425–466, 2008.

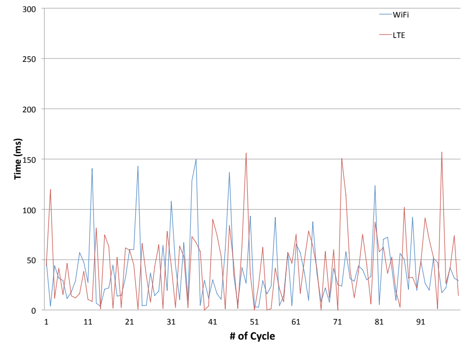

Fig. 14. A comparison of the arrival times.

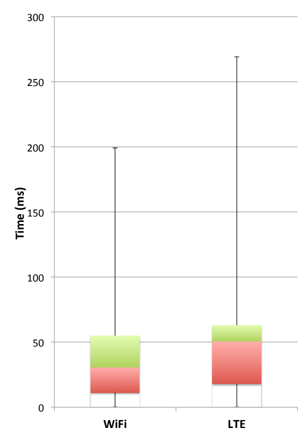

Fig. 15. Summarized box plotting of the arrival times.

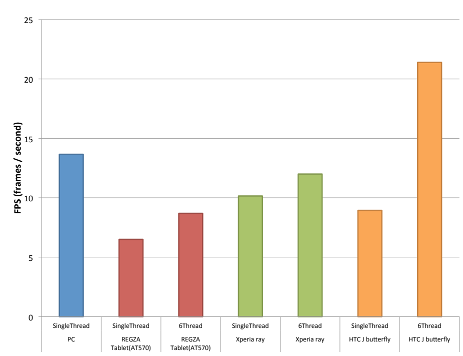

Fig. 16. Performance differences of a smartphone and PC.