A Context Synchronization Method to Reduce Offloading Network Overhead for IoT-Cloud Virtual Machine System

Junho Jeong¹, Yunsik Son² and Yangsun Lee³

¹Electronic Commerce Institute, Dongguk University
123 Dongdae-ro, Gyeongju 38066, Korea
yanyenli@dongguk.edu
²Dept. of Computer Engineering, Dongguk University
30 Phildong-ro 1gil, Jung-Gu, Seoul 04620, Korea
sonbug@dongguk.edu
³Dept. of Computer Engineering, Seokyeong University
124 Seokyeong-ro, Sungbuk-Ku, Seoul 02713, Korea
Corresponding Author: yslee@skuniv.ac.kr

Abstract. In this paper, we have studied a context synchronization technique that reduces network communication overhead by extracting contexts that require synchronization through static profiling. The context synchronization scheme of this paper can reduce the network communication overhead because it tries to synchronize only the context necessary for offloading execution.

Keywords: Internet of Things, Cloud System, Offloading, Virtual Machine, Context Synchronization, Static Profiler

1 Introduction

The IoT-Cloud virtual machine system [1] provides high-performance cloud server computing power to the Internet of Things (IoT) equipment using offloading techniques. When the offloading is executed, context synchronization is required because the execution program must be consistent between the IoT device and the cloud server to execute given applications normally. Existing system context synchronization attempts full context synchronization when offloading is executed. The existing context synchronization technique is inefficient because the network communication overhead in context synchronization is proportional to the size of the context to be synchronized. In this paper, we introduce a context synchronization technique that reduces network communication overhead by extracting contexts that require synchronization through static profiling.
2 IoT-Cloud Virtual Machine System

The IoT-Cloud VM system can accommodate all contents written in C / C++, Java and JavaScript languages by applying the advantages of a platform-independent environment of the existing smart virtual machine [1]. In addition, the offloading technique [2] can be applied to the cloud server environment to perform tasks requiring high-performance computing power on low-performance IoT devices. Figure 1 shows the offloading module structure of the IoT-Cloud VM system.

![Fig. 1. System Configuration of the IoT-Cloud VM system.](image)

The Static Profiler analyzes the performance and data flow of the application in advance by searching the target application or metadata without actually executing the program. In the IoT-Cloud virtual machine system, the static profiler analyzes the intermediate language, SIL (Smart Intermediate Language), and creates a control flow graph by grouping the instructions in blocks in the control flow graph generator.

The CFGA (Control Flow Graph Analyzer) analyzes each pattern represented by the generated control flow graph and generates profile information by predicting the performance of the application based on a predefined complex metric.
3 Context Synchronization Method

Context information that needs to be synchronized is divided into essential context information that is unconditionally required for offloading execution and stack context information in which operations and variables are managed. In this paper, IoT-Cloud virtual machine determines synchronization scope of stack contexts based on context information extracted through static profiling and attempts to synchronize only necessary areas.

The operation stack synchronization uses transfer parameter information and return value information extracted through static profiling because it must be tried when there is a transfer parameter or a return value. If the parameter or return value is not an address value, synchronization is attempted as much as the data size of the entire parameter or return value in the frame area of the offloading execution object function. If the value is an address value, it converts the absolute address value stored in the operation stack to a relative address value, and then attempts to synchronize the frame region of the execution target function.

The activation record synchronization uses address parameter information extracted through static profiling because synchronization should be attempted when address value of local variable is parameter. Since the address value of the local variable refers to a value managed in a frame area other than the frame area of the offloading target function, only the area to be referred to must be synchronized. Therefore, the reference area of the active record is calculated by using the data size information of the parameter and the relative address, and only the area required for the offloading is tried to be synchronized.

4 Conclusion

The IoT-Cloud virtual machine system uses an offloading technique that provides high-performance computing power to low-performance devices. In this case, context synchronization is required because the program to be offloaded must maintain consistency between the server and the client. The network overhead caused by context synchronization may reduce the performance of offloading execution, and a method for reducing network overhead is needed.

In this paper, we study the static profiling-based context synchronization technique to reduce the network overhead cost of context synchronization of existing systems. Since the context synchronization scheme of this paper synchronizes only the context information necessary for offloading execution, the network overhead cost is reduced, and efficient offloading is possible compared with the existing system which tries to synchronize the whole context.
Acknowledgments. This research was supported by Basic Science Research Program through the National Research Foundation of Korea(NRF) funded by the Ministry of Science, ICT and Future Planning (No. 2016R1A2B4008392). And, this research was supported by Basic Science Research Program through the National Research Foundation of Korea(NRF) funded by the Ministry of Education(NRF-2017R1D1A3B03029906).

References