Caching Mechanisms towards Single-Level Storage Systems for Internet of Things

Yosuke Bando[1], Konosuke Watanabe[1], Ken-ichi Maeda[1], Hiroki Kudo[2], Masahiro Ishiyama[2], Atsushi Kunimatsu[1], Hiroto Nakai[1], Masafumi Takahashi[1], and Yukihito Oowaki[1]

[1]Semiconductor & Storage Products Company, Toshiba Corporation, Yokohama, Japan
[2]Corporate Research & Development Center, Toshiba Corporation, Kawasaki, Japan
yosuke1.bando@toshiba.co.jp

Abstract

Internet of Things (IoT) involves coping with enormous number of distributed devices. This paper introduces three pieces of caching technology as steps towards single-level storage systems that can host and map numerous IoT devices on a single vast address space: 1) a caching mechanism for making solid-state storage appear as huge main memory, 2) speeding up access to resource-limited IoT devices by caching the address translation table of solid-state storage chips, and 3) ad hoc device-to-device data relay, which can be used as effective network caching for mapping IoT devices.

Introduction

In the coming era of Internet of Things (IoT), storage systems must cope with enormous number of distributed devices connected primarily through wireless networks. We envision single-level storage systems [1] that can logically host and map numerous IoT devices on a single vast address space, where user programs can access those devices without having to be aware of the actual storage types or physical device locations, as shown in Fig. 1. We introduce three pieces of caching technology for such IoT-oriented storage systems as small but concrete steps towards the aforementioned envisioned goal.

Single-Level Storage Systems based on Flash Memory

Using NAND flash memory suitable for typically small and mobile IoT devices, we implemented two proof-of-concept, partial prototypes of a single-level storage system, where data can be accessed in a unified way regardless of whether it is on the DRAM or on the flash memory. To realize this, a storage memory management unit (SMMU) is introduced as shown in Fig. 2 to cache flash memory data on the DRAM, making the flash memory appear as huge main memory that can be accessed via load/store instructions rather than read/write system calls. This is similar to virtual memory and memory-mapped IO, but our goal here is to use flash memory as main memory, and the main challenge is to hide limitations of flash memory such as a limited number of write cycles. Our first prototype is a virtual machine (VM) where the guest VM's main memory is mapped to the host VM's flash memory. A software SMMU on the host VM allows the guest VM to access data on the flash memory as if it were on the main memory. The SMMU tries not to write cached DRAM data back to the flash memory and compresses it when write-back is necessary, eliminating 82% of write-back traffic for video playback as shown in Fig. 3. Our second prototype is a hardware implementation of the dotted rectangle part in Fig. 2. A Linux® boot sequence successfully runs on it, demonstrating the feasibility of the system.

Low-Latency Flash Memory Access using Host DRAM

Even with the use of DRAM cache, it is still desirable to have faster flash memory. The access latency of solid-state storage devices comes from the fact that they translate requested addresses into device internal addresses in order to reduce the number of erase operations by writing to different parts of the storage. Since the address translation table is usually stored in a portion of the flash memory itself, called flash translation layer (FTL), FTL access is required in addition to actual data access. One solution to reducing the FTL access latency is to add DRAM cache to the storage controller chip, but this increases the chip size and power consumption, which is unsuitable for resource-limited IoT devices. Instead, since we consider host devices that are equipped with an SMMU, we propose to cache the FTL table on the host DRAM and to have the SMMU perform this additional address translation, as shown in Fig. 4. Our storage chip that can access the host FTL [2] reduces the access latency from 270 μsec to 133 μsec.

Wireless Ad Hoc Device-to-Device Data Relay

Network caching is needed for mapping remote IoT devices on a single address space. However, it is unlikely that wireless network infrastructures alone will be able to satisfy growing mobile data traffic. We are developing a system for ad hoc device-to-device data relay [3], where each device caches data, so that other nearby devices can access it via proximity-based wireless communication (e.g., Wi-Fi®) without going through preexisting networks, as shown in Fig. 5. As a first step, we consider presumably the simplest scenario of one IoT device delivering data to others in the same area. The challenge here is to efficiently transfer data by avoiding radio interference in the absence of central control, which we address by using multicast transmission augmented with a packet repair mechanism, along with adaptive selection of relay nodes according to signal levels of nearby devices [4]. Simulation indicates that our method delivers 128kB data to 7000 devices in 50 sec, which is seven times faster than naïve unicast flooding. We equipped our prototype device with a FlashAir™ Wi-Fi SD card with modified firmware as a wireless network adapter, to have more control over the multicast bitrate and transmission power than typical off-the-shelf adapters in order to optimize performance.

Conclusions

Three caching mechanisms have been demonstrated through individual prototypes, which we believe to be integral parts for IoT-oriented single-level storage systems. Emerging nonvolatile memory such as MRAM can certainly give a performance boost, whose study will be an interesting future direction.

Acknowledgments

Special thanks to Henry Holtzman and V. Michael Bove, Jr. for their valuable advice; to Tsutomu Owa and Takashi Omizo for their help in developing the single-level storage prototypes.

References

[1] J. S. Shapiro et al. USENIX Tech Conf, 59-72, 2002.
[2] K. Watanabe et al. ISSCC, 330-331, 2014.
[3] D. Dubois et al. ESEC/FSE, 687-690, 2013.
[4] T. Sakoda et al. IEICE Tech Rep 114(417), 101-106, 2015.

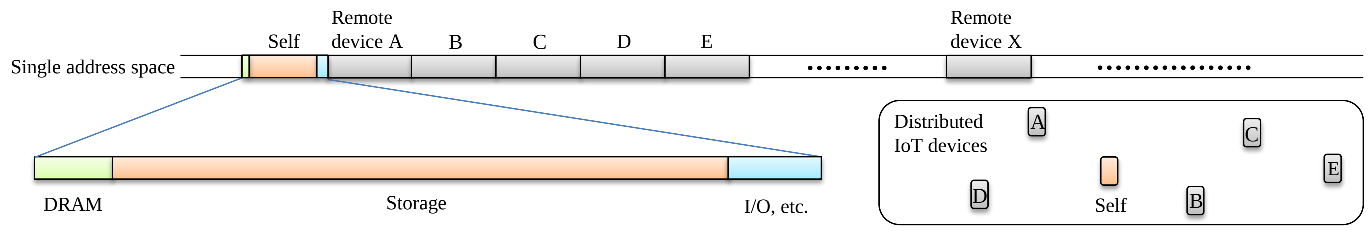
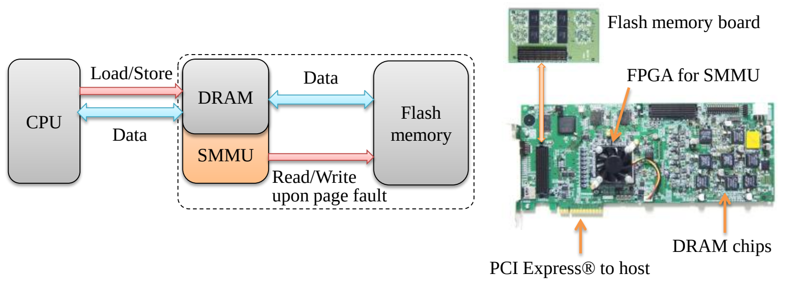


Figure 1. Single-level storage system where distributed IoT devices are mapped onto a single huge address space.



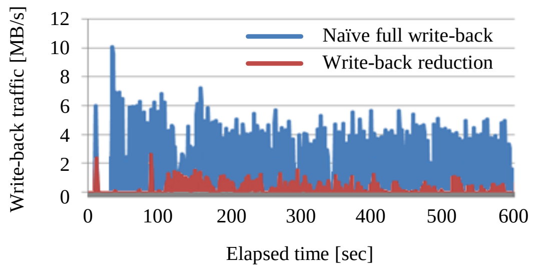


Figure 2. Proposed storage system allowing for load/store access to the flash memory with the aid of a storage memory management unit (SMMU), making the flash memory appear as huge main memory, along with the prototype board corresponding to the dotted rectangle part.

Figure 3. Plots of the amount of data traffic being written from the DRAM back to the flash memory as video playback progresses on the virtual machine prototype.

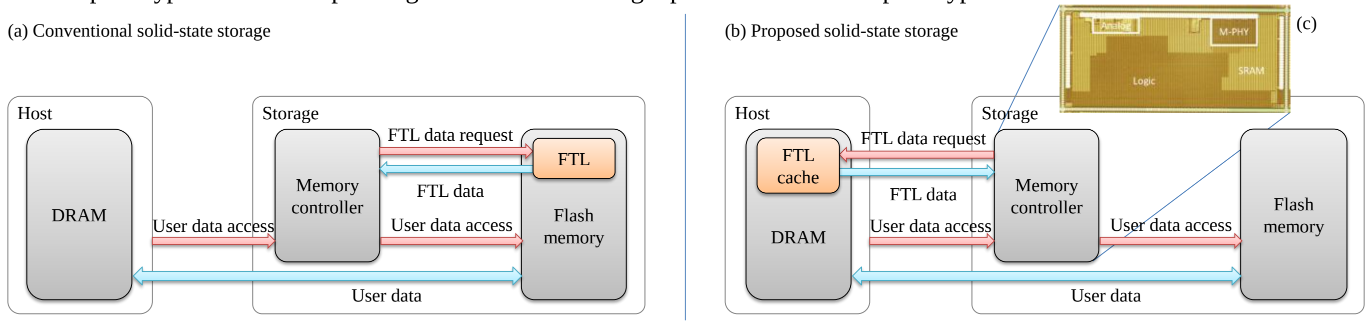


Figure 4. (a) Conventional solid-state storage needing to first read address translation information stored in the flash translation layer (FTL) in the flash memory before accessing user data, resulting in an increased latency. (b) Proposed storage controller eliminating this additional flash memory access by caching FTL data in the host DRAM. (c) Micrograph of the prototype chip.

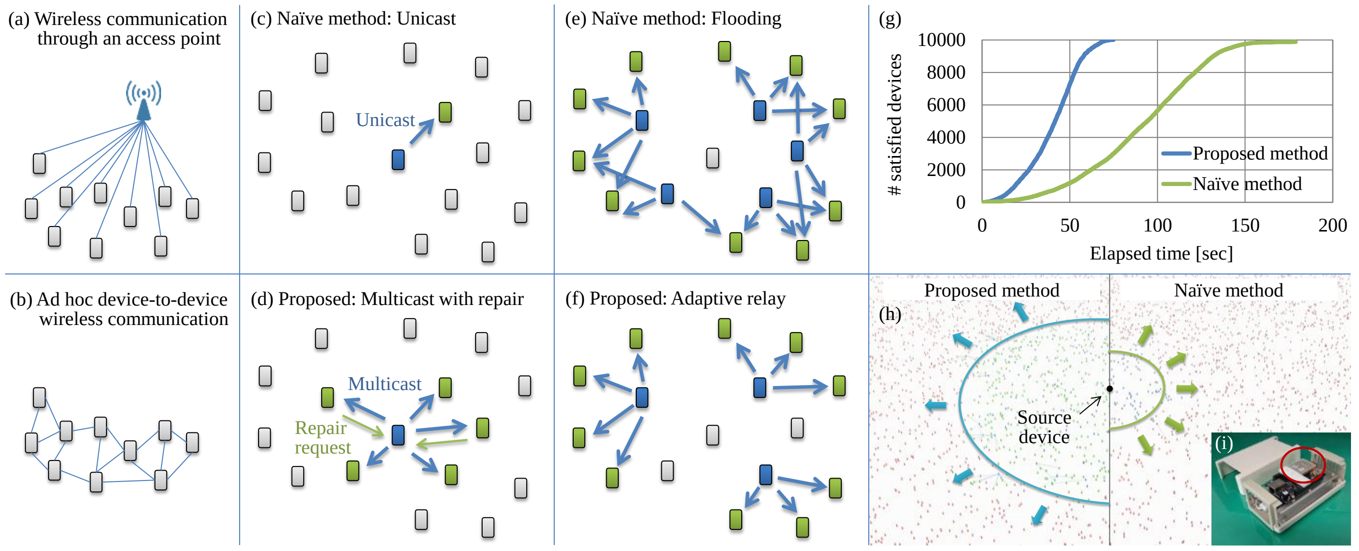

(a) Wireless communication through an access point
(c) Naïve method: Unicast
Unicast
(e) Naïve method: Flooding
(g)
# satisfied devices
Proposed method
Naïve method
Elapsed time [sec]
(b) Ad hoc device-to-device wireless communication
(d) Proposed: Multicast with repair
Multicast
Repair request
(f) Proposed: Adaptive relay
(h)
Proposed method
Naïve method
Source device
(i)

Figure 5. (a) Distributed devices connected via a preexisting wireless network. (b) Devices connected directly with each other nearby, where, as opposed to mesh networks, data is transferred in a store-and-forward way so that each device acts as network cache. (c) Unicast transmission where a sender (blue) delivers data to one receiver (green) at a time. (d) Multicast transmission where surrounding receivers are served simultaneously and can request repair for lost packets. (e) Flooding where all the devices that have received the data relay it, causing interference. (f) Adaptive relay where only selected devices relay the data. (g) Plots of data delivery progress in a simulation of 10,000 devices receiving data from a single source device. (h) Simulation snapshot where green/blue dots indicate devices that have received the data, while red dots indicate those that have not. (i) Prototype device equipped with a FlashAir™ Wi-Fi SD card as indicated by the red circle.

FlashAir is a trademark of Toshiba Corporation.