Technical Disclosure Commons

Defensive Publications Series

April 2020

Cloud-Based Virtual Memory System in which Memory Pages Are Offloaded to Cloud Storage

Robert Benea

Patrik Torstensson

Rajeev Kumar

Cloud-Based Virtual Memory System in which Memory Pages Are Offloaded to Cloud Storage

Abstract:

This publication describes systems and techniques to implement virtual memory to swap memory regions, such as memory pages, between a resource-constrained electronic device and cloud storage. Many "always-connected" or internet of things (IoT) devices are resource constrained such that dynamic random-access memory (DRAM) and flash memory are significantly limited. Excessive write operations can also reduce a usable lifetime of the flash memory. For cloud-based virtual memory, pages corresponding to working sets of more important applications are maintained in DRAM at the device. In contrast, a virtual memory system moves memory pages that are used less frequently based on some metric (e.g., least recently used (LRU)) from the DRAM to cloud storage that is located at a remote server. The virtual memory system may compress the memory pages to save transmission and/or storage bandwidth. The remote pages may also be updated without moving them back to the device. In this manner, an electronic device can have more or larger active applications without over-consuming DRAM or increasing wear on the flash memory.

Keywords:

resource-constrained, always-connected device, internet of things (IoT), virtual memory, memory page, swap, cloud storage, ram disk, zram, compression, compressed page, cold page, least recently used (LRU), offload

Background:

Many electronic devices have significantly lower resources than notebook computers or even smartphones. Examples of resource-constrained devices include television streaming devices, smart speakers, personal voice assistant devices, and IoT devices. In addition to a less powerful processor, such devices typically have relatively smaller volatile and non-volatile memories. For volatile memory, these devices usually employ DRAM. For non-volatile memory, such devices often rely on flash memory. Due to size and cost constraints, both the DRAM and the flash memory of these types of devices are limited to such a degree that it can be difficult to have multiple applications running simultaneously, or even just a few larger applications.

When an application is running, the data that the application is accessing—called the "working set"—is stored somewhere on the electronic device. Applications are executed faster if the working set is stored in DRAM instead of flash memory. Unfortunately, when multiple applications are running, the memory consumed by the total working set across multiple applications can exceed the size of the DRAM. To enable the multiple applications to still be running at the same time, virtual memory techniques are used.

Virtual memory enables the electronic device to store more working set data than the DRAM can ordinarily do on its own. To do so, a virtual memory space is created by "pretending" that the DRAM is larger than it physically is and by dividing the virtual memory space into regions, which are often called memory pages. For this virtual memory space to work, data that exceeds the actual physical size of the DRAM is stored elsewhere or in a special manner. In one approach, this additional virtual memory storage space is carved out of the flash memory. A memory page that contains overflow information is moved from the DRAM to the flash memory to free space in the DRAM. Here, overflow information corresponds to information that exceeds the physical size

of the DRAM and is unlikely to be used again soon. The overflow information is thus stored in the flash memory with this approach. Accessing the flash memory with multiple store cycles, however, increases wear on the flash memory, and flash memory has a limited lifetime in terms of storage cycles. Relying solely on flash memory to enable virtual memory can, therefore, reduce the usable lifespan of the electronic device.

In another approach, the DRAM itself is used to provide additional virtual memory storage space beyond the baseline physical specification of the DRAM by employing a special storage technique. Specifically, using the DRAM for expanded virtual memory space can be enabled via a special storage technique that involves data compression. A memory page with overflow information is compressed such that a given quantity of data consumes less physical storage in the DRAM after the compression. For example, a four kilobyte (4 KB) memory page may be compressed to 1.5 KB. From an available memory space perspective, compressing a memory page within the DRAM is equivalent to moving a portion of the memory page out of the DRAM. Although this second approach prevents additional wear on the flash memory, each memory page that is "paged out" of the DRAM still occupies a portion of the DRAM. Accordingly, this approach is still ultimately limited by the size of the DRAM, which is particularly impactful for resource-constrained devices having only a small quantity of DRAM.

Therefore, it is desirable to provide techniques that enable resource-constrained devices, including those that use flash memory for non-volatile storage, to use virtual memory to expand a number of applications that can be simultaneously running or to increase a complexity of applications that can be executed.

Description:

Described virtual memory techniques enable memory pages to be offloaded from an electronic device to cloud storage. A cloud-based virtual memory system can transfer a cold page, which has not been accessed recently, from the device to a server over a network, such as the internet. When the offloaded page is needed locally again, the electronic device retrieves the memory page from the cloud storage for use at the device. Alternatively, the virtual memory system can update the offloaded page by sending a packet to the server providing the cloud storage. In some cases, the cold page is compressed before being offloaded to the cloud storage. This compression can reduce both transmission and storage bandwidth. Compression can also reduce the size of an update packet.

This cloud-based virtual memory system can also be used in conjunction with paging memory to local flash memory and/or paging memory to DRAM via compression. Selection of a virtual memory approach for a given memory page can be based, for example, on a likelihood of how soon the memory page will be accessed again or whether a delay in accessing a page can affect a critical task of the device.

An example cloud-based virtual memory system is described with reference to Fig. 1. An example cloud-based virtual memory technique is described with reference to Fig. 2.

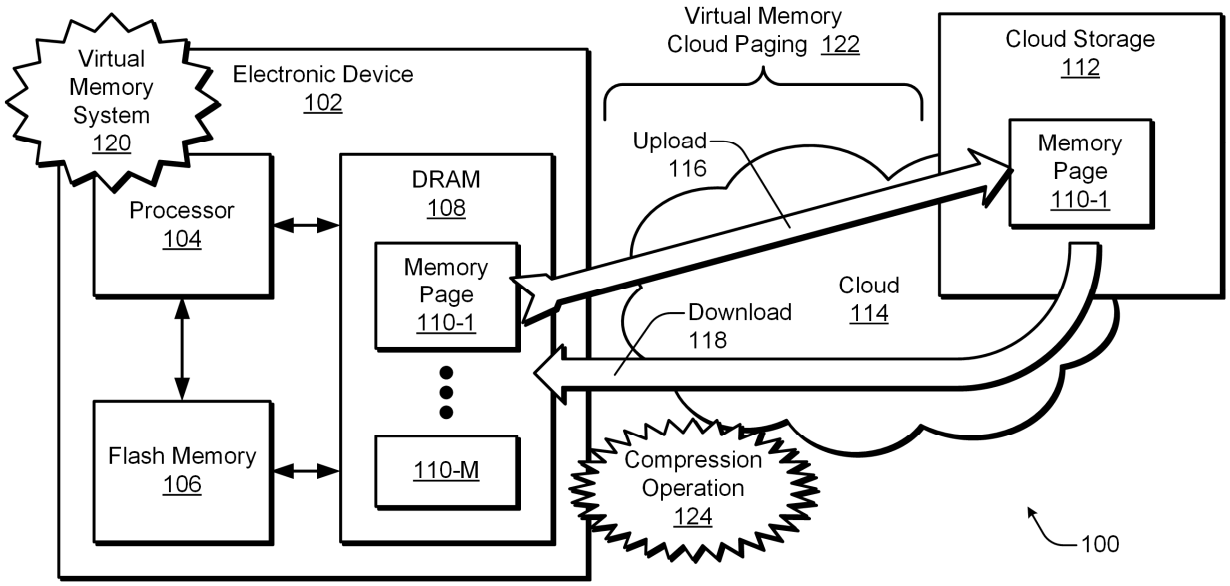

Fig. 1. Virtual memory system that offloads pages to cloud storage.

As shown in Fig. 1, a virtual memory environment 100 is distributed between an electronic device 102 and cloud storage 112. The electronic device 102 includes at least one processor 104, at least one flash memory 106, and at least one DRAM 108. Each of the processor 104, the flash memory 106, and the DRAM 108 can be coupled one to another. Alternatively, the DRAM 108 can be implemented as a main memory that is directly accessible by the processor 104, and the flash memory 106 can be implemented as backing storage that is accessed through the DRAM 108. Other computing architectures can be implemented instead.

The processor 104 realizes a virtual memory system 120 that separates the virtual memory space into multiple memory regions. Examples of memory regions include equal-sized memory pages and variable-sized memory segments. For clarity, the memory regions are referred to as memory pages in the examples presented herein, but the described principles are applicable to other types of memory regions. As shown, the virtual memory is divided into multiple memory pages 110-1…110-M, with "M" representing a positive integer. Although not explicitly depicted in Fig. 1, a memory page 110 can also be paged into the flash memory 106 and/or into a compressed

area of the DRAM 108. Using data compression or creating a compressed area of a DRAM for virtual memory is also referred to as "zram," "compcache," or using a RAM disk for virtual memory.

The virtual memory environment 100 includes at least one network, which is represented by a cloud 114. The network may include the internet, a wireless local area network (WLAN) (e.g., a Wi-Fi network), a wireless wide area network (WWAN) (e.g., a cellular network), combinations thereof, and so forth. The electronic device 102 is coupled to the cloud storage 112 via the cloud 114. The cloud storage 112 can be implemented with at least one server, such as an internet server farm including data storage capabilities. Examples of virtual memory cloud paging 122 are described next.

In example operations, the virtual memory system 120 offloads a memory page 110 using virtual memory cloud paging 122. The virtual memory cloud paging 122 includes an upload operation 116 and a download operation 118. As part of the virtual memory cloud paging 122, the virtual memory system 120 determines a candidate memory page 110 for offloading from time to time. Any of many different techniques for determining a candidate memory page 110 can be employed. Example techniques involve an LRU analysis, identifying unused pages over some time interval, detecting that a maximum memory utilization percentage has been reached, discovering that another memory page is to be paged back onto the device, and combinations thereof.

Responsive to a determination that a memory page 110-1 is to be paged off the electronic device 102, the virtual memory system 120 implements virtual memory cloud paging 122. For example, to perform an upload operation 116, the virtual memory system 120 causes the electronic device 102 to transfer the memory page 110-1 to the cloud storage 112 via the cloud 114. Prior to

or as part of the upload operation 116, the electronic device 102 can compress the memory page 110-1 to reduce a size thereof as part of a compression operation 124. Accordingly, the electronic device 102 can transfer a compressed version of the memory page 110-1 to the cloud storage 112.

The cloud storage 112 is therefore responsible for retaining the memory page 110-1. During the page retention period, the electronic device 102 may decide to update the memory page 110-1. To do so, the virtual memory system 120 can transfer a complete copy of the updated memory page 110-1 to the cloud storage 112 as a replacement. If, for example, a 4 KB page is compressed to around 1.5 KB, a single internet protocol (IP) packet may be sufficient to replace the retained copy of the memory page 110-1 at the cloud storage 112. Alternatively, the virtual memory system 120 may transmit an indication of the changed portions of the memory page 110-1 while omitting unchanged portions.

At some point, the electronic device 102 may decide that it is necessary or more efficient to execute some application while the memory page 110-1 is stored locally. To return the memory page 110-1 to the DRAM 108, the virtual memory system 120 causes the electronic device 102 to perform the download operation 118. The download operation 118 includes retrieving the memory page 110-1 from the cloud storage 112 for storing in the DRAM 108. If the compression operation 124 was performed for the upload operation 116, the electronic device 102 can decompress the memory page 110-1 and store a decompressed version of the memory page 110-1 in the DRAM 108. Alternatively, the electronic device 102 can store a compressed version of the memory page 110-1 in the DRAM 108—e.g., like a zram implementation to conserve space in the DRAM 108.

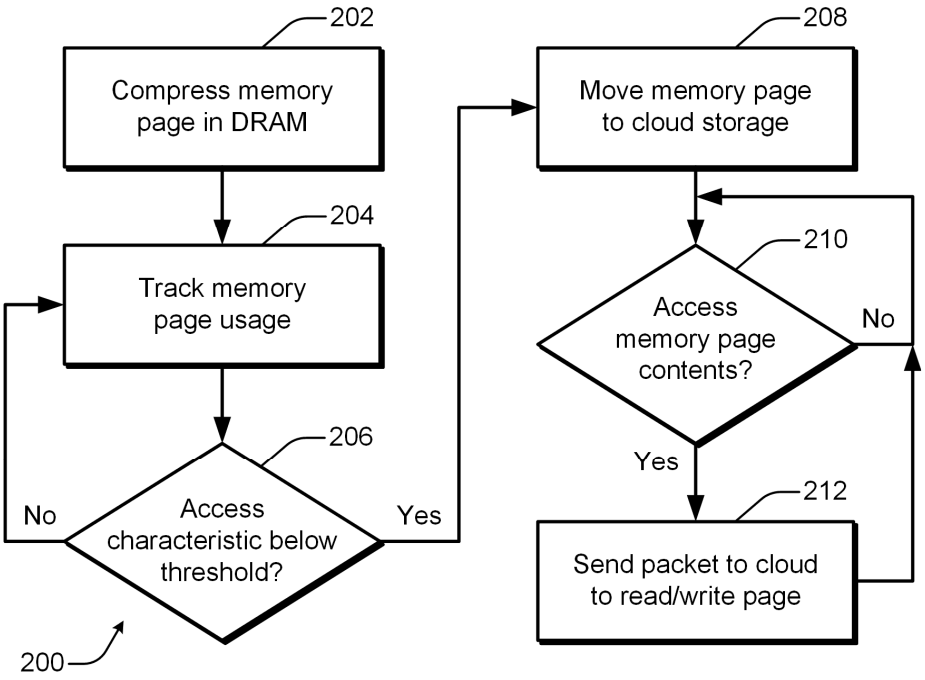

Fig. 2. Process for operating a cloud-based virtual memory system.

Fig. 2 depicts an example technique 200 to operate a cloud-based virtual memory system. The technique 200 includes six (6) blocks 202-212. At block 202, a memory page 110 is compressed and stored in the DRAM 108 in accordance with a compressed-block RAM procedure, e.g., using zram principles. Alternatively, the memory page 110 may remain uncompressed to reduce demands on the processor 104. At block 204, the virtual memory system 120 tracks memory page usage in accordance with a scheme to trigger memory paging. For example, each memory page 110 may be flagged, and a timer may be started. Each time a memory page 110 is accessed, the corresponding indicator is unflagged.

At block 206, the virtual memory system 120 determines if an access characteristic is below a threshold. Continuing with the page-flagging example, any memory page 110 that remains flagged upon expiration of the timer has an access characteristic that is below the threshold. If the access characteristic of certain memory pages 110 is not below the threshold, then the technique 200 continues at block 204 by tracking usage for those memory pages. On the other hand, if the

access characteristic of a given memory page 110 is below the threshold, then the technique 200 transitions to block 208 for that selected memory page 110-1. Other types of thresholds or virtual memory criteria may alternatively be used. For instance, any memory page 110 that is not accessed at least twice over a two-day period may be a candidate for offloading. Also, candidates may be selected for offloading based on having a lowest relative access frequency over some time period, such as an hour or a week.

At block 208, the virtual memory system 120 moves the selected memory page 110-1 to the cloud storage 112 using a network of the cloud 114 as part of an upload operation 116. While the memory page 110-1 is retained at the cloud storage 112, the electronic device 102 may determine at block 210 whether an application is attempting to access the contents of the memory page 110-1. If not, the virtual memory system 120 may refrain from any activity with that memory page 110-1 until an access does occur at block 210. If, however, the electronic device 102 does access the memory page 110-1 at block 210, the virtual memory system 120 generates at least one packet indicating an access operation. The access operation can be a read or a write operation. If the access operation is a write, the packet can include full replacement data or just changes for the memory page 110-1. At block 212, the virtual memory system 120 sends the generated packet to the cloud storage 112 to read or alter at least a portion of the memory page 110-1. If an access frequency of a particular memory page 110-1 exceeds a threshold, the virtual memory system 120 can return the particular memory page 110-1 to the electronic device 102 as part of a download operation 118.

References:

[1] Patent Publication: US20180364918A1. Managed NAND Cold Data Storage. Priority Date: June 19, 2017.

[2] A. Boroumand et al. 2018. Google Workloads for Consumer Devices: Mitigating Data Movement Bottlenecks. In ASPLOS '18: 2018 Architectural Support for Programming Languages and Operating Systems, March 24–28, 2018, Williamsburg, VA, USA. ACM, New York, NY, USA, 16 pages. http://dx.doi.org/10.1145/3173162.3173177.