SCIENCE CHINA
Information Sciences

• LETTER •

July 2020, Vol. 63 179101:1–179101:3
https://doi.org/10.1007/s11432-019-2672-5

IMCI: an efficient fingerprint retrieval approach based on 3D stacked memory

Wen CHENG[1], Ran CAI[1], Lingfang ZENG[1*], Dan FENG[1], André BRINKMANN[2] & Yang WANG[3*]

[1]*Wuhan National Laboratory for Optoelectronics, School of Computer Science and Technology, Huazhong University of Science and Technology, Wuhan* 430074, *China;*
[2]*Department of Computer Science, Johannes Gutenberg-University Mainz, Mainz* 55128, *Germany;*
[3]*Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences, Shenzhen* 518055, *China*

Received 11 June 2019/Revised 2 August 2019/Accepted 2 September 2019/Published online 17 February 2020

Citation Cheng W, Cai R, Zeng L F, et al. IMCI: an efficient fingerprint retrieval approach based on 3D stacked memory. Sci China Inf Sci, 2020, 63(7): 179101, https://doi.org/10.1007/s11432-019-2672-5

Dear editor,

Data deduplication (Dedup for short) as a type of redundant data elimination technology, can effectively reduce the impact of redundant data on storage costs, which consequently alleviates the problems and pressures caused by massive data storage, management, and backup. As such, with the exponential growth of user data stored in storage systems, data deduplication systems are gaining increasing popularity in diverse applications.

Data deduplication typically consists of several steps: data segmentation, fingerprint calculation, and fingerprint retrieval. In practice, the fingerprints of data blocks, which are often calculated by using SHA-1 or MD5 secure hash digest algorithms, are used to represent and identify identical data blocks, while fingerprint retrieval is responsible for determining whether a data block is duplicated. Although fingerprint calculation may incur high overhead because of the complexity of fingerprint algorithms, the overhead of fingerprint retrieval is much worse for the data deduplication process [1] because, given the large volume of data stored in deduplication systems, the fingerprints of existing blocks must be repeatedly retrieved for comparison with high frequency, which could quickly become a performance bottleneck in deduplication processing. As such, optimizing fingerprint search is a great challenge that must be addressed.

Existing fingerprint-search schemes mostly focus on mining data characteristics to design new index-access strategies or using high-performance storage devices to speed up the fingerprint retrieval [2, 3]. For example, considering the large number of fingerprint indexes generated by mass data, some indexes are stored in hard disks. By taking advantage of data locality, the index data hit ratio in memory is improved, thereby reducing access to the slow hard disk and speeding up the fingerprint retrieval [2, 3]. However, given the separation of calculation and storage, a fingerprint must be continuously moved between the central processing unit (CPU) and memory through the memory bus, leading to higher time overhead and energy consumption, which affects the speed of fingerprint retrieval.

Traditional computer memory is generally built using DRAM (dynamic random access memory), and the limited bandwidth of DRAM is a key performance bottleneck in a modern computer system. The memory bandwidth issue becomes increasingly severe as the number of CPU cores increases and memory-intensive applications become more popular. The bandwidth of traditional memory limits the efficiency of fingerprint transmission. Therefore, because of technical and hardware constraints, no fast and energy-saving fingerprint re-

* Corresponding author (email: lfzeng@hust.edu.cn, yang.wang1@siat.ac.cn)

trieval scheme has been widely used thus far.

Three-dimensional (3D) stacked memory is an emerging storage technology, which is internally connected by through-silicon via technology (TSV technology) [4], featuring high capacity, high bandwidth, and low power consumption. To speed up data processing and release CPU potential, hardware architects propose an "in-memory processing" solution that allows memory to provide high bandwidth and low latency.

This study introduces a new and efficient fingerprint-retrieval approach based on 3D stacked memory, called in-memory chunk identification (IMCI). IMCI utilizes PIM (processing in memory) to move the logic operation of fingerprint retrieval into memory in order to reduce the interaction between the CPU and memory and avoid unnecessary data movements, which consequently improves the retrieval efficiency and reduces energy consumption.

We exploit the Rabin algorithm to determine the data block breakpoint and then use the SHA-1 algorithm to process each data block to obtain its 160-bit fingerprint, which is used to compare the stored hash values indexed via a hash table that maps the fingerprints of the stored blocks to their physical address. In IMCI, the data block is duplicate if a match is found, otherwise, the fingerprint of the data block is unique. It is then stored and indexed via the hash table, and the corresponding data blocks are aggregated into fixed-sized containers (typically 4 MB). In performing fingerprint retrieval, a stack cube modeled by Cascaded-IO (SMLA_CIO) is accessed in IMCI via a crossbar network by the CPU.

IMCI assumes SMLA_CIO as its memory, which is a 4-layer memory stack. The SMLA divides the entire stacked cube vertically into logical channels, similar to the concept of Vault in the hybrid memory cube [5,6]. These channels can work in parallel.

In this design, the SHA-1 [7] algorithm is used to process the data block to obtain a 160-bit hash value as its fingerprint. If the fingerprint is regarded as a hexadecimal string, then each fingerprint has 40 bits (where a "bit" is defined as the number of characters in the string). The first character of the fingerprint is one of '0'–'9' or 'A'–'F', which is also called the flag of the fingerprint. Given the nature of SHA-1, the number of fingerprints with each different flag would be very close; hence, we consider that the data block fingerprint can be classified into 16 categories according to its flag.

For these 16 flags of the data block fingerprints, we set the SMLA to support 16 vertical logical channels to utilize its parallelism. The data block fingerprints are then evenly distributed throughout the 4 stacked memory layers. Therefore, the channel can process 16 data block fingerprints in parallel. The SMLA is evenly divided into 16 areas, and each of these areas stores the fingerprints with the corresponding flag.

The used 3D stacked memory contains a logic layer, in which the computational units can be embedded to equip the processing logic of a particular application. As mentioned earlier, the SMLA is divided into 16 channels. A process element (PE) is also set for each channel to manage the vertical partitions and perform the fingerprint search operations. Thus, the logical layer has a total of 16 PEs. Taking the previous example, the flag of fingerprint 'h' is '3', and should be stored in the Channel3 partition.

The fingerprint determines the storage partition according to its flag. When a fingerprint arrives, it must be forwarded to the corresponding PE for processing. The router is responsible for the communication between the CPU and logical layer. After the CPU sends fingerprint 'h' to the router, the router forwards it to PE3 according to the flag of 'h'. After completing the retrieval, PE3 returns the result to the router, which transmits it to the CPU.

Two queues (i.e., a request queue and a result queue) are set on the router and used to forward the request to a specific PE and return the result to the CPU from the specific PE. A global buffer is also set on the routing chip to temporarily store the data block fingerprint. A simple calculation unit is configured to process the forwarding logic.

To determine if a fingerprint exists in the hash table, IMCI first uses the hash function (the hash function here is an algorithm, such as SHA-1 or MD5) to calculate the fingerprint, then determines its index position in the hash table, and finally goes to the location to read the data and compare it with the fingerprint. A match indicates that the fingerprint already exists and the data block is a duplicated data block not requiring any further processing; otherwise, the fingerprint does not exist and the data block is a new one. However, if the fingerprint of this data block has a conflict in the storage location in the hash table, the fingerprint data must be written to a new storage location via calculation.

Each PE in IMCI processes its own fingerprint. 16 PEs can process 16 data block fingerprints in parallel without interfering with each other because the communication between PEs is not necessary, thereby saving the communication overhead of the logic layer. Each PE sends a corresponding operation command (read/write) accord-

ing to the comparison results. Depending on the command type, the memory controller then performs the corresponding operations on the storage layer. A buffer is set in each PE to temporarily store the data block fingerprint. Each PE allocates a task according to the flag, and when consecutive identical flags appear, the request queue needs to wait. IMCI maintains the request queue for data block fingerprints in each PE and adopts the FCFS (first come first serve) policy to process the fingerprints.

IMCI adopts a hardware configuration similar to that of Neurocube [8]. The PE unit uses a 28 nm manufacturing process. The area budget of each PE is 0.1936 mm^2, and 16 PEs occupy a total of approximately 3.09 mm^2. The area budget of a router is originally 0.0609 mm^2. The area cost of the router is assumed to be the same as that of the PE, and the budget of the entire logic layer is 68 mm^2. Therefore, IMCI can be integrated into the logic layer.

IMCI stores the data block fingerprints in its memory. Data will be lost after power-off because the memory is volatile. The data persistence scheme can be used to write the indexes to the hard disks. IMCI pays attention to the data block fingerprint retrieval stage. In alleviating the unnecessary interference, the data index is assumed to be completely stored in the memory without data exchanges between the upper and lower storage devices. IMCI's fingerprint retrieval process begins after the CPU calculates the data block fingerprint:

(1) The CPU sends the fingerprint to the router of the SMLA logical layer, and the router controls the next operation;

(2) The router forwards the data to the corresponding PE according to the flag of the data block fingerprint;

(3) The PE first inserts the data block fingerprint into its own request queue and then takes the fingerprint 'h' from the head of the queue and sends it to the operator and comparator;

(4) The operator takes the fingerprint as the key and performs a hash calculation to obtain its position index in the hash table;

(5) The operator sends the address information to the memory controller;

(6) The memory controller reads the data of the address from the DRAM, puts it into the buffer, and sends it to the comparator for comparison with 'h';

(7) The PE determines the type of command based on the comparison result and sends the command type to the memory controller;

(8) The memory controller controls the DRAM to perform the read/write operations.

Conclusion. In this study, we proposed a new and efficient fingerprint-retrieval approach based on 3D stacked memory, called IMCI, which adopts the PIM idea by migrating the deduplicated logic to the internal memory, thereby avoiding the data movement overhead for fingerprint calculations. IMCI selects the SMLA_CIO-type 3D stacked memory as the storage medium and designs the partition and logical layer of the storage layer separately. With these improvements, the system throughput of fingerprint retrieval can be increased by 12.24%–26.64%, and the average storage energy consumption is reduced by 22.95%.

However, the 3D stacked memory in IMCI is volatile memory, and the data will be lost if the power is turned off. Therefore, the data must periodically persist on the hard disks. This process can result in significant time and energy overhead. Thus, well-designed persistence methods to minimize the overhead or new non-volatile 3D stacked memory to store the block fingerprints are required. This is our next research direction.

Acknowledgements This work was partially supported by National Natural Science Foundation of China (Grant Nos. 61821003, 61672513, 61572377) and Science and Technology Planning Project of Guangdong Province (Grant No. 2019B010137002).

References

1 Kim J, Lee C, Lee S, et al. Deduplication in SSDs: model and quantitative analysis. In: Proceedings of the 28th Symposium on Mass Storage Systems and Technologies (MSST), 2012
2 Zhu B, Li K, Patterson H. Avoiding the disk bottleneck in the data domain deduplication file system. In: Proceedings of the 6th USENIX Conference on File and Storage Technologies, 2008
3 Lillibridge M, Eshghi K, Bhagwat D, et al. Sparse indexing: large scale, inline deduplication using sampling and locality. In: Proceedings of the 7th Conference on File and Storage Technologies, 2009. 111–123
4 JEDEC Standard. High bandwidth memory (HBM) DRAM. JESD235, 2013. https://www.jedec.org/standards-documents/docs/jesd235a
5 Pawlowski J T. Hybrid memory cube (HMC). In: Proceedings of IEEE Hot Chips 23 Symposium (HCS), 2011
6 Jeddeloh J, Keeth B. Hybrid memory cube new dram architecture increases density and performance. In: Proceedings of Symposium on VLSI Technology (VLSIT), 2012. 87–88
7 Eastlake D, Jones P. Us Secure Hash Algorithm 1 (SHA1). Technical Report, 2001
8 Kim D, Kung J, Chai S, et al. Neurocube: a programmable digital neuromorphic architecture with high-density 3D memory. In: Proceedings of the 43rd Annual International Symposium on Computer Architecture (ISCA), 2016. 380–392