Mapping Block Information Recovery

Alahmadi Abdulhadi*, Tae Sun Chung**
*Dept of Computer Engineering, Ajou University
e-mail:obd-7@hotmail.com

Conclusion

Our scheme going to use the advanteges of shadow technical and use it in flash memory. With our scheme we can maintain the system from corruption by making a capy table for table mappinng. It is make us to recover the data easily when something unusually happen to the system.

Abstract: In this paper we proposed Mapping Block Information Recovery (MBIR) for flash memory based storage devices. According to the data access problems in mapping method our idea can save the map block information in two tables , the first one is orginal and the scond one is a capy one. We can do it by using shadow technical. We going to using shadow technical advantages to be familiar with Flash Translation Layer (FTL) to improving the capacity for flash memory .

1- Introduction

Flash memory is a constantly-powered nonvolatile semi-conductor device that has the advantages of small size, fast access, low power consumption, convenient portability, shock resistance, data retention after a power off, random access, and heat dissipation. Flash memory is currently being integrated with different embedded system devices such as in digital cameras, smart phones, PDAs, and sensor devices[1,2,5].

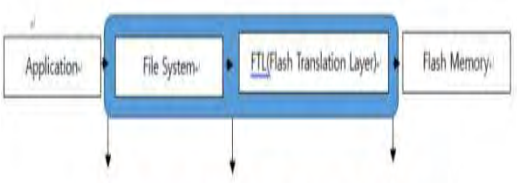
The read/write/erase behavior of flash memory is radically different than that of other programmable memories, such as volatile RAM and magnetic disks. Perhaps more importantly, memory cells in a flash device (as well as in other types of EEPROM memory) can be written to only a limited number of times.

The rminder of this paper is organized as follows: Section two, describe related work. Section Three, describe Mapping Block Information Recovery Scheme.

2- Related work

2.1. Flash Translation Layer (FTL):

Fig. 1 shows the software architecture of file system as following: Application, file system , flash translation layer and flash memory. Our idee a is focuses on the FTL layer shown in (Fig 1).



(Fig 1) Architecture of flash memory system

FTL emulates disk-like in-place updates for a Logical Page Number (LPN) based data page as follows: FTL first acquires a clean page and updates the corresponding data on that page. Then, it maps the original LPN into this new Physical Page Number (PPN). Thus,

an efficient FTL scheme has a critical effect on overall performance of flash memory since it directly affects in-place update performance and balancing the wearing of each data block[1,5].

2.2.FTL characteristics

2.2.1 An FTL algorithms should be consider the following characteristics:

The firest one, is the main logical-to-physical address mapping the main job for it is to convert logical addresses between the file system and physical address in flash memory[2].The second one, we should to condider power-off recovery to preserved data structure and guarantee data consistency.Finally, to maintain earase operation should include wear-leveling technical.

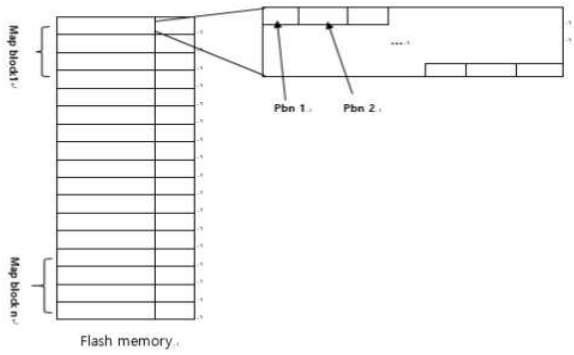
2.3 Mapping adress and maping information

Its very important to cosider a scheme to store mapping information. To be able to recavery mapping table when a power-on process events.Mapping information loss should be able to recavery in the sudden power-off events.Because of that this information must be saving in somewhwere of flash memory.The storing mapping information in flash memory can be devided to two classes: the map block method and per block method.

2.3.1 Map block method

A map block method put mapping information in dedicated blocks of flash memory termed map blocks. Most FTL implementations use more than one map block. (Fig 2) shows the way of map blocks how store the mapping information[1].

If mapping information updated or changes, the above recording job have to be done. During performing the recording job, if the map blocks pool have no unused sector, erase technology have to be moving to another free



(Fig 2) Map block method

map blocks.For fast loockups we can cashed the mapping table in RAM[3].

So, we have to rebuilt the mapping table in RAM during a power-on process.

3.Mapping Block Information Recovery Scheme (MBIR)

In our proposed scheme the idea is to make another capy for mapping information table. We can use the capy table to recover the old mapping information, when tha data have some updating or changes. The update or changing in the map teable will make the system corruption[3]. Our scheme whit using shadow technical can solve this problem and recover the mapping table information. The basic idea is to use the shadow thcnichal advantages and make it familiar whith algorithms of FTL. Our goals are to add the shadow algorithms to FTL stornig algorithm to be able to recover the mapping information

REFERNCES

[1] T.-S. Chung, D.-J. Park, S. Park, D.-H. Lee, S.-W. Lee, and H.-J. Song, "A survey of Flash Translation Layer," *J. Syst. Archit.*, vol. 55, no. 5-6, 2009.

[2] E. Gal and S. Toledo, "Algorithms and Data Structures for Flash Memories," in *ACM Computing Surveys*, vol. 37, no. 2, 2005.

[3] Se Jin Kwon, Tea-Sun Chung, An efficient and advanced apace-managmenet technique for flash memory using reallocation blocks, IEEE Transaction on Consumer Electronics 48 (2) (2002).

[4]Samsung Electronics, Nand flash memory and Smartmedia, Data Book,2007.

[5] Intel Corporation Understanding the Flash Translation Layer (FTL) Specification.

[6] Eran Gal, Siven Toledo, Mapping Structure for Flash Memories:Techniques and Open Problems. SWSTE '05 Proceedings of the IEEE International Conference on Software - Science, Technology & Engineering Pages 83 - 92, 2005.