Design and Implementation of cache protecting from power failure in Disk Array

WANG Endong, HU Leijun,LV Shuo, CHENG Jichen, WEN Zhongling, ZHANG feng
(Mass storage research Department,state key laborator y of high-end server&storage technology, beijing 100085)

Abstract

In high-end disk arrays, the ups supplied the power to controllers, however, not to the JBOD. In this paper, a solution to cache protecting was designed and implemented by the method of software. In this solution, the controllers redirected the dirty page of the cache to the reserved partition of the system disk as soon as the commercial power failure was detected. The data protected in the system disk partition was recovered to the JBOD when the commercial power became normal. So the cache data was not lost after power failure. It was proved that the data in the cache can be exactly recovered when the commercial power failure happened to the storage system.

Keywords: disk array; cache; Uninterruptible Power Supply; data redirection

1. Introduction

The speed of processor increased much faster than that of memory[1], and the area of memory is commonly tended to focus on a smaller continuous area when the main memory was accessed[2]. So from these two points of view, a buffer called cache was inserted between the processor and the main memory. The page cache was designed in the linux kernel to decrease the hard disk IO operation[3]. Specifically, the IO operation from upper layer was put into memory firstly instead of directly into the slow hard disk.

In order to improve the sequential read and random write performance of the high-end disk array, the cache is applied to ours' storage controller. A special memory is deviled from the whole main memory. This special memory is used for the high-speed cache of the JBOD(Just a Bunch Of Disk). The cache improved the performance of the storage system, however, the memory used for cache are volatile. That is to say, the date in the cache will be lost forever when the power supply off or fails. The date is very important to the enterprise special for the applies of OLTP. What's more, the operate system will not recognize so that this will lead to statistics difference between upper layer and lower layer of the system. So, it will make very serious loss to the customers or users.

UPS(Uninterruptible Power System) is one of the device which can make the voltage and frequency of power supply stabled, and it's main function is to supply the power continually when the commercial power off[4]. It's principle is as follows: when the commercial power works normally, the UPS works as a voltage stabilizer, and also charge the battery inside. When the commercial fails, the UPS immediately converted power of the battery to alternating current to the load to maintain the normal operation. In the existing high-end disk array, the UPS can only supply power to the controller, rather than the back-end JBOD, therefore, a software cache power-down protection program is designed in this paper. When the commercial power is down, the cache data of the controller were written to a special disk partition on the controller system disk. As the system power is restored , restore the saved data to the back-end JBOD .

2. Design

2.1 System architecture

When the power supply works normally, the user data is written to the controller cache; The cache module took a certain policy to flush the data in the cache to the back-end disk array. When the commercial power supply is down, the controller is supplied with the UPS. The back-end disk array is power-down, and the UPS power supply time is limited, so we write all the cache data in all controllers into the special disk. In our design, the special disk partition is the reserved system partition, and the data saved before will be written to the back-end disk array when commercial power is recovered. The design idea of cache protection system is showed in Figure 1.

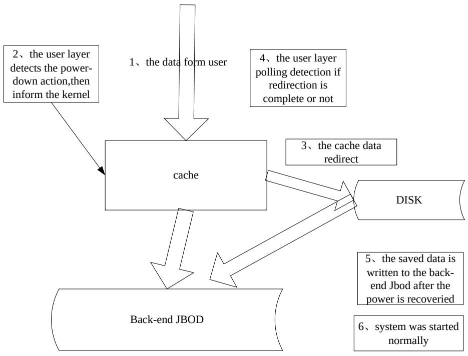
2、the user layer detects the power-down action,then inform the kernel
1、the data form user
4、the user layer polling detection if redirection is complete or not
cache
3、the cache data redirect
DISK
5、the saved data is written to the back-end Jbod after the power is recoveried
Back-end JBOD
6、system was started normally

Figure 1 The design ideas of cache protection system

The software architecture of the cache protection system is showed in Figure 2. In our architecture, the dirty pages searching module is mainly responsible for the searching the dirty pages in page cache, the error page IO processing module is mainly responsible for processing IO error page of back-end JBOD caused by the power-down, the page frame recycling module is responsible for the recovery of the cache page data, and the Pdflush thread module is mainly responsible for writing data to the specified disk.

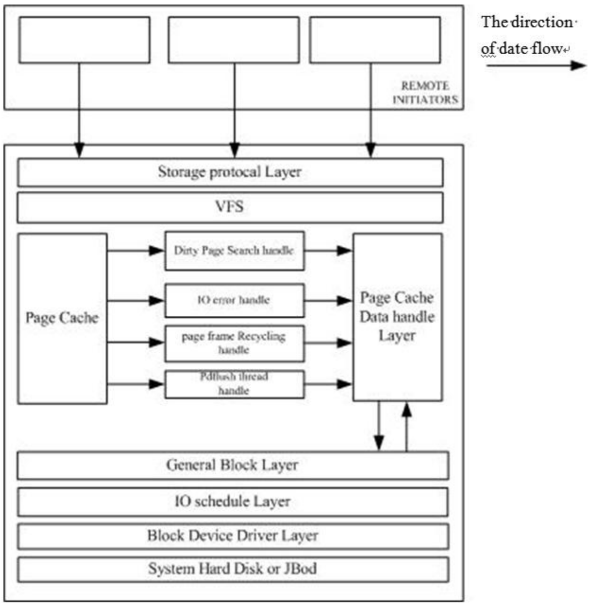

Fig.2 The software architecture of the cache protection program

2.2 The overall process of the system

The controller received IO requests sent by the client data through exchange network. When the controllers find the power is down, the UPS sends a message to the host manager module, then it makes cache protecting module work for data redirection. The cache data protection process should be devilled into two steps: Firstly, dirty data in the cache is written to the specified disk, and secondly, the cache data saved in the special disk partition is read and recovered to back-end JBOD. The stage two can be finished in user space, however, the stage one must be done in kernel state. Overall, the process can be recognized as data redirection. The overall process is shown in Figure 3 .

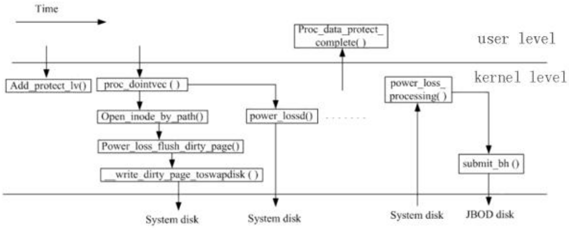

Fig. 3 the overall process of cache protection

As we know, the kernel runs concurrently. When the manager module detests the power down signal, the kernel sets a flag that indicates power down. In this process, If the kernel is submitting a written request to the JBOD but has not yet returned to the kernel, the thread for pages flushing or pages recycle may be scheduled for running. As a result , the pages write error, So in this case, we use a list to save the pages marked the error flag. And within the kernel timer, make a judgment that whether has the necessarily to write the pages to the special disk partition. If needed, then modify the block address and offset of “bio” to write the “bio” to the appointed partition; if not needed, then submit an IO error report.

The kernel thread for searching the dirty pages will be started to search dirty pages belonged to the back-end JBOD after detecting power down. The dirty pages are written to appointed disk partition according to mapping relationship.

The Pdflush thread and memory recycle thread are response for data flushing in Linux kernel[5]. We distinguish and set a flag to the address_space object belonging to the back-end JBOD, and then stop writing to the back-end JBOD.

Two methods are adopted to promise the data integrity: firstly, as soon as the commercial power is down, the kernel is informed by the “proc/sys/vm/upsup” interface, the information indicates that the cache data should be protected; secondly, IO processing callback function must identify error flag and then add the error page to the specific list to deal with the case that the kernel is submitting IO request but does not get response form the back-end JBOD.

2.3 Kernel informed by the message of power-down

The cache protecting module runs in kernel status, but the thread that detects the power-down signal runs in user status, an interface from the user to kernel layer is needed by the message to pass through. Here we use the proc files system to realize the interface. For specifics, we add “ups” to “vm_table” in the file /kernel/sysctl.c. As soon as detecting that power is down, the manager module inform the cache protecting module through the interface. The source code is shown as follows.

```
{
     .ctl_name          = CTL_UNNUMBERED,
     .procname          = " upsup ",
     .data              = &sysctl_upsup,
     .maxlen            = sizeof(sysctl_upsup),
     .mode              = 0644,
     .proc_handler      = &proc_dointvec,
     . .strategy        = &sysctl_intvec,
     }
```

2.4 IO processed of the error page

The error page Written to the back-end JBOD will trigger the completion of processing of the IO page callback function. Usually, the error IO message may be printed because of the error pager. In our program, if the page is belonged to the back end disk , the page will be, added to the specified list we defined before, and then the page will marked dirty, next, take the traversal the other buffer_head in the page. If it is asynchronous writing, then unlock returned, otherwise judge if none of the page in the asynchronous writing state, then unlock, clear page writeback mark , wake up the process of waiting for the page write back .

In our implementation, the error page is added to head_power_loss list which define by our own. When the linked list is not empty, error page processing thread is waken up, and the page descriptor is get from the list , and the page management area is acquired. timeout detection is used by the timer to distinguish between IO errors caused by the power-down and other IO errors. Protection is given up if IO error not due to power-down; the first buffer head of this page is get and the page fault mask is cleared. Then all the buffer head is checked whether have IO error, if it is error, redirected to the specified system disk partition.

2.5 Search and processing of dirty pages

The dirty pages search process start as soon as the identification of power-down is detected, Parts of dirty pages to disk belong to controllers’ system partition , and some dirty pages belongs to the backend JBOD. In this program, the dirty pages are distinguish by address_space which gained by opening the device node. The dirty pages that belongs to the back end JBOD are written to the specified system disk partition according to the mapping relationship. The way by which to redirect the dirty pages and error IO page are almost the same, just have to change device number and sector number.

2.6 The process of metadata writing

The data block should be written to the partition device, additionally, the description the data block and its description such as the device number of the data block on the back-end storage devices, LBA(Logical Block Addressing) and length, should also be written to the partition device. In order to avoid writing metadata to disk whenever the data block is written, metadata information should be written only when its quantity reaches a certain level instead of being written to the disk each time, aiming to improve the overall performance. Metadata information should be written to the cache every time the data block is written, which includes the index of write data on the disk, device number and sector. Metadata information will be returned when its quantity reaches a certain level, then the times for writing the superblock in the cache should be updated (mark the flag of the superblock as dirty). The remaining metadata information will be flashed to the disk at a time when all the data is written to the disk, meanwhile the superblock information on the cache will be flashed to the disk.

2.7 Process for recovering data

The system is restarted after power is restored, then make a decision whether or not to write data by reading the flag of the superblock in the partition of system disk. The data recovery process should be started if the flag means the system is powered off. Firstly, the description information of the page is read, which keeps a record of which position on the disk the corresponding data block should be written to; Secondly, the page offset is calculated according to the index of the descriptor. In order to search the corresponding file pointer for writing, a linked list should be saved, which records the correspondence between the block device and the file, because there are more than one back-end disk will be written. Thirdly, the corresponding page data is read form system partition, then the data will be written to the specified position on the JBOD according to the device number of the descriptor and the index on the block device. Finally, when all the page data is flushed into the back-end JBOD, the flag of the superblock is cleared and another flag is set, which indicates the recovering is finished. A kind of transformation is mainly performed here, the block address on partition will be transformed into the one in the back-end JBOD. The process of writing data to the back-end JBOD is illustrated in figure 7.

3 System test

What's important is that the data in the cache can be stably and reliably protected in the case of power break by applying cache power-down protection, meanwhile, the data was stably and accurately recovered after the power is restored. Therefore, test was as follows.

The hardware configuration of the test system was as follows.

Controller: CPU: E5620 *2; memory: 16GB; system disk: 500GB; JBOD: 500GJ; SAS card: LSI 3801E; network card: gigabit NIC; UPS: Delta GES-N7K.

Client-server: CPU: E5504; memory: 6GB; network card: gigabit NIC; operating system: Redhat Enterprise 5.4 x86_64; services protocol: iscsi.

After copying all the different size of data from or to the disk, which mapped from the ISCSI device (first check the MD5 value of the source data), we removed Commercial power immediately. The data was written to the specified partition on the system disk by the power-down protection module, till the commercial power was recovered. We verified the MD5 value of the recovered data and compared it with the one of the source data. The result of the experiment is shown in Table 1. From Table 1, we can see the cache data at the controller can be well protected by the cache power-down protection module in the case of power break, and the MD5 value of the recovered data is equal to the one of the source data. Therefore, the consistency of the data can be ensured.

Table 1 the result of the test

items	size	The MD5 of source data	The MD5 of data removed
1	4K	620f0b67a91f7f74151bc5be745b7110	620f0b67a91f7f74151bc5be745b7110
2	4M	b5cfa9d6c8febd618f91ac2843d50a1c	b5cfa9d6c8febd618f91ac2843d50a1c
3	40M	ec8bb3b24d5b0f1b5bdf8c8f0f541ee6	ec8bb3b24d5b0f1b5bdf8c8f0f541ee6
4	400M	61eabaf2bf278703738b433ff884c91f	61eabaf2bf278703738b433ff884c91f
5	4000M	ffe3915bd77fde9dd5dc8077ced09c10	ffe3915bd77fde9dd5dc8077ced09c10

4. Conclusion

In this paper, a technical solution based on UPS cache power-down protection is proposed and implemented to solve the problem that the cache data at the controller will be lost in the case of power break. The result indicates that the cache data can be stably and reliably protected by the cache power-down protection in the case of power break, and the recovered data is same to the source data. It is meaningful for this cache power-down protection to improve the reliability and the availability of the storage service.

5 References

[1] Li Ming, Tang Zhi-min. Partial cache locality: A new approach to cache optimization[J]. Chinese Journal of Computers, 1997, 20(1): 1-8

[2] Jiang Yi, Jiang Jun-jun, Xiong An-ping. Collaborative cache consistency strategy based on object-based storage file system[J]. Computer Engineering and Design, 2012,33(11):4204-4208

[3] Bovet, Cesati. Understanding the Linux Kernel.3rd[M].New York：OREILLY, 2007

[4] Shi Lei. Development of UPS technology[J]. Electric Switchgear, 2009, 01: 8-10

[5] Kroab-Hartman,Corbet, Rubin. Linux Device Driver.2rd[M]. New York:OREILLY, 2001