The Journal of Systems and Software 82 (2009) 1298–1312

Contents lists available at ScienceDirect

The Journal of Systems and Software

journal homepage: www.elsevier.com/locate/jss

FAST: Flash-aware external sorting for mobile database systems ☆

Hyoungmin Park, Kyuseok Shim *

School of Electrical Engineering and Computer Science, Seoul National University, Kwanak, P.O. Box 34, Seoul, Republic of Korea

A R T I C L E I N F O

Article history:
Received 25 June 2008
Received in revised form 29 December 2008
Accepted 24 February 2009
Available online 12 March 2009

Keywords:
Flash memory
External sorting
Mobile database system

A B S T R A C T

Recently, flash memory has gained its popularity as storage on wide spectrum of computing devices such as cellular phones, digital cameras, digital audio players and PDAs. The integration of high-density flash memory has been accelerated twice every year for past few years. As flash memory's capacity increases and its price drops, it is expected that flash memory will be more competitive with magnetic disk drives. Therefore, it is desirable to adapt disk-based algorithms to take advantage of the flash memory technology.

In this paper, we propose a novel Flash-Aware external SorTing algorithm, FAST, that overcomes the limitation of larger writing cost for flash memory to improve both overall execution time and response time. In FAST, we reduce the write operations with additional read operations. We provide the analysis for both traditional and our flash-aware algorithms by comparing the detailed cost formulas. Experimental results with synthetic and real-life data sets show that FAST can result in faster execution time as well as smaller response time than traditional external sorting algorithms.

© 2009 Elsevier Inc. All rights reserved.

1. Introduction

Recently, flash memory has been gaining its popularity as storage on wide spectrum of emerging computing devices such as cellular phones, digital cameras, digital audio players and PDAs. Flash memory offers many advantages of better energy efficiency, higher throughput for read operations, compact size, no noise, better shock resistance and low latency data transfer than magnetic hard disks (Douglis et al., 1994). The main disadvantages of flash memory are higher price and the requirement of erase operation before updating existing data.

Current flash memory technologies offer two quite different types: NAND and NOR. NOR flash memory has long erase and write times, but allows random-access to any memory location. This makes it a suitable replacement for older ROMs which are used to store program codes such as a computer's BIOS that rarely needs to be updated.

In contrast, NAND flash memory provides faster erase and write times, better storage densities, up to ten times of the endurance and lower costs. However, the I/O interface of NAND flash does not provide a random-access to a memory location, but offers block-wise read only with typical block sizes of hundreds of bytes. Furthermore, it offers much slower read operation than NOR flash memory. This makes NAND flash similar to other secondary storage such as hard disks that are suitable for use in mass-storage devices.

To speed up write operations of NAND flash memory, a flash storage called SSD (solid state drive) is developed. SSD consists of several NAND flash chips. When writing a number of pages, several pages are written at a time to multiple chips to reduce overall write time.While SSD has improved write speed, read operation is still slow due to the nature of NAND type.

Recently, to speed up read operation, a hybrid version of NAND and NOR types called OneNAND[1] is developed and is available in the markets. It takes both advantages from high speed data read of NOR type's interface logic and the advanced data storage function of NAND type. Thus, it has faster reading than NAND flash memory. However, the gap between write and read costs is large. For currently available OneNAND flash memories, a sequential page write is about 10 times slower than a sequential page read. (See more details in OneNAND[1].)

The integration of high-density flash memory has been accelerated twice every year for past few years (Kimura and Kobayashi, 2003). It is expected that this trend will continue and hundreds of gigabytes of flash memory will be available in the near future. As its capacity increases and price drops, flash memory will be more competitive with magnetic disk drives (Paulson, 2005). Thus, it is not surprising to hear that one of the leading flash memory manufacturer recently announced the production of new laptop computers without magnetic disks.

☆ This work was supported by the IT R&D program of MIC/IITA [2006-S-040-01, Development of Flash Memory-based Embedded Multimedia Software].

* Corresponding author.
E-mail addresses: hmpark@kdd.snu.ac.kr (H. Park), shim@ee.snu.ac.kr (K. Shim).

[1] Refer to http://www.samsung.com/global/business/semiconductor/products/fusionmemory/products_onenand.html.

As flash memory is replacing magnetic disk drives as an alternative storage device, it is natural to observe a database management system running on light weight computing devices with flash memory. For example, consider a phone book data, where a user may conduct query operations with sort-based algorithms such as searching the phone number of the closest restaurant (top-k query), navigating the names in the alphabetical order to find the phone number of a relative that the user cannot remember his/her name and so on. However, in such mobile computing environments, the users are not patient with waiting for the result of interactive querying. Thus, the database management systems running in the mobile devices must be designed to respond as quickly as possible with the first few results while they complete the processing of the entire query efficiently. Therefore, it is desirable to adapt the database management system softwares to take advantage of the flash memory technologies in order to bring the application softwares running with the database management systems in such mobile products faster response and overall execution times.

It was reported from a leading mobile phone manufacturer that the available main memory allowed for database systems is very limited (e.g. 2 MBytes) because several applications in a mobile device must compete each other for small main memory in the mobile device. Furthermore, the competition between RAM and other resources such as CPU and flash memory on the same silicon die used in Systems on Chip (SoC) (NRC Report, 2001) restricts the main memory capacity in the lightweight computing devices. Therefore, main memory still remains as one of the critical resources in mobile environments and developing database algorithms such as external sorting in such environments to work with flash memory effectively for both response time and overall execution time is very critical.

With a software layer called Flash Translation Layer (FTL) (Intel, 1998), which allows flash memory to be accessed as if it is hard disk, the disk-based algorithms used in the traditional database systems work without any modification. On the other hand, it is not tuned for the best performance by utilizing the asymmetric costs of read and write operations in the flash memory for mobile database systems including those in commercial products and research prototypes. For example, depending on flash memory's type (e.g. NAND, NOR and OneNAND), manufacture and model, the asymmetric cost ratio of write and read operations significantly varies. Thus, it is likely to provide suboptimal performance in mobile database systems where the flash memory with a high cost ratio of sequential write and read is used. These unique characteristics of flash memory necessitate elaborate flash-aware data structures and algorithms in order to effectively utilize flash memory as data storage media.

In this paper, we present FAST, a novel flash-aware external sorting algorithm that overcomes the limitation of larger writing costs for flash memory to improve both overall execution time and response time. In FAST, to alleviate the high latency of sequential write operations, we reduce the write operations with the introduction of additional read operations. Whenever write operations are much more expensive than read operations in the flash memories used in the mobile database systems, FAST can improve the performance of sorting significantly.

FAST has the following novel combinations of characteristics that effectively capture flash memory characteristics and requirements of mobile environment:

- **Fast execution of external sorting:** The traditional disk-based external sorting algorithms are not optimized to reduce the writing cost. However, FAST is designed to reduce the write operations with the introduction of additional cheap read operations for the flash memories equipped with more expensive writing than reading.
- **Fast response time of external sorting:** One of important requirements in mobile computing devices is to get a few results as early as possible. Thus, to achieve fast response time, the query processors in commercial embedded mobile database systems utilize pipelines as much as possible. FAST is also designed to speed up the execution time without much sacrifice of response time for external sorting.
- **Long life of flash memories:** Since flash memory has a finite number of erase cycles, reducing the number of write operations based on response time requirement will also prolong the useful life of flash memory.

As we will show later, traditional disk-based style external sorting performs well on a flash memory with similar costs of sequential page read and write, but performance suffers when the flash memory has a high cost ratio of sequential write and read. We provide the detailed analysis of I/O cost for both traditional and our algorithms so that the best external sorting algorithm can be selected by query optimizer depending on the characteristics of flash memory available in the mobile database systems.

We also implemented our FAST and conducted an extensive experimental study with synthetic and real-life data sets on both NAND and NOR type flash memory boards, with more expensive write costs, that are available in the markets. Experimental results show that FAST has not only faster execution time but also smaller response time than traditional external sorting algorithms. Furthermore, if we are willing to tolerate the execution time to speed up the response time, we could achieve even smaller response time with FAST.

2. Related work

The problem of sorting has been studied extensively in computer science area because of wide applications in real-life. The fundamental external sorting algorithms including replacement selection and merge sort are extensively covered in Knuth's book (Knuth, 1973). Since the fundamental external sorting algorithms were introduced, many researchers investigated the sorting problem in order to improve the performance of them in various circumstances (Anciaux et al., 2003; Larson, 2003; Larson and Graefe, 1998; Nyberg et al., 1994, 1995; Pang et al., 1993; Sinha and Zobel, 2003; Yiannis and Zobel, 2007). Vitter surveyed external sorting algorithms in Vitter (2001) for large date sets that do not fit entirely in main memory focusing on optimizing I/O costs of large seek time in disk.

While the speed of processors increases at a greater rate than that of both disk and memory, cache-aware external sorting algorithms have been studied in Larson (2003), Nyberg et al. (1994, 1995) and Sinha and Zobel (2003). These approaches consider main memory and cache as if hard disk and main memory, respectively. Then, they reduce memory access costs such as cache misses by adopting similar technique of reducing disk I/O for the external sorting algorithms.

Larson (2003) and Larson and Graefe (1998) extended the external sorting algorithm with replacement selection, which works only for fixed length record data sets, to be able to process the data sets with variable length records. They developed a new memory management technique to handle the variable length records.

External sorting of a large data set involves many I/O operations. Recently, Yiannis and Zobel (2007) proposed a new method which compresses intermediate run files to reduce both the number of runs and the size of runs for external sorting. Experimental result shows that I/O cost of external sorting is improved.

Anciaux et al. (2003) proposed a sorting algorithm under the very limited resources including main memory. Thus, no

Download English Version:

https://daneshyari.com/en/article/461573

Download Persian Version:

https://daneshyari.com/article/461573

Daneshyari.com