Information Sciences 327 (2016) 183–200

Contents lists available at ScienceDirect

Information Sciences

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

In-storage processing of database scans and joins

Sungchan Kim [a,*], Hyunok Oh [b], Chanik Park [c], Sangyeun Cho [c], Sang-Won Lee [d], Bongki Moon [e]

[a] Chonbuk National University, Republic of Korea
[b] Hanyang University, Republic of Korea
[c] Samsung Electronics Co., Ltd., Republic of Korea
[d] Sungkyunkwan University, Republic of Korea
[e] Seoul National University, Republic of Korea

ARTICLE INFO

Article history:
Received 15 September 2014
Revised 3 March 2015
Accepted 29 July 2015
Available online 22 August 2015

Keywords:
SSD
Database
Performance
Energy
Scan
Join

ABSTRACT

Flash memory-based SSD is becoming popular because of its outstanding performance compared to conventional magnetic disk drives. Today, SSDs are essentially a block device attached to a legacy host interface. As a result, the system I/O bus remains a bottleneck, and the abundant flash memory bandwidth and the computing capabilities of SSD are largely untapped. In this paper, we propose to accelerate key database operations, scan and join, for large-scale data analysis by moving data-intensive processing from the host CPU to inside flash SSDs ("*in-storage processing*"), close to the data source itself. To realize the idea of in-storage processing in a cost-effective manner, we deploy special-purpose compute modules using the System-on-Chip technology. While data from flash memory are transferred, a target database operation is applied to the data stream on the fly without any delay. This reduces the amount of data to transfer to the host drastically, and in turn, ensures all components along the data path in an SSD are utilized in a balanced way. Our experimental results show that in-storage processing outperforms conventional processing with a host CPU by over up to 7 ×, 5 × and 47 × for scan, join, and their combination, respectively. It also turns out that in-storage processing can be realized at only 1% of the total SSD cost, while offering sizable energy savings of up to 45 × compared to host processing.

1. Introduction

Recently, there is a substantial influx of NAND flash-based Solid State Drives (SSDs) in the enterprise storage market [18]. For example, large-scale storage appliances like Teradata's Extreme Performance Appliance [42] and Oracle's Exadata [33] harness SSDs to realize very high I/O operations per second (IOPS). Moreover, there are a host of positive forecasts for SSDs in the enterprise market [31].

Most prior developments have focused on the potential of flash SSD as fast and cost-effective hard disk (in terms of IOPS/$) with the legacy block interface. Taking into account the impressive performance and other advantages of flash SSD, this approach is very practical and will remain mainstream for a while. However, we observe that this conventional usage model of SSD would

* Corresponding author. Tel.: +82-63-270-2411.
E-mail addresses: sungchan.kim@chonbuk.ac.kr (S. Kim), hoh@hanyang.ac.kr (H. Oh), ci.park@samsung.com (C. Park), sangyeun.cho@samsung.com (S. Cho), wonlee@ece.skku.ac.kr (S.-W. Lee), bkmoon@snu.ac.kr (B. Moon).

http://dx.doi.org/10.1016/j.ins.2015.07.056
0020-0255/

fail to fully exploit the performance improvement opportunities offered by continued technology advances. Notably, the aggregate raw bandwidth of flash memory devices in SSDs has already exceeded the peak bandwidth of most existing legacy host interfaces. Furthermore, future SSD controllers are expected to have high computing horsepower by necessity, as they integrate parallel flash interfaces, large high-speed memory, and more computing cores and logic. Maintaining the legacy block devices implies, unfortunately, that the abundant flash memory bandwidth and the computing capabilities inside an SSD will be largely untapped.

Meanwhile, new digital data are generated at astounding rates. Digital information in corporations, on the public Internet and on home computers is doubling every month [16].

The enterprise systems domain is no exception (e.g., eBay's 2.4 PB relational data [34]). Large-scale data analysis has become a necessity and will be increasingly important for enterprises. In order to efficiently warehouse and analyze data at such scales, a shared-nothing, massive parallel processing storage tier of many storage servers is common [48].

Unfortunately, the performance of data-intensive applications would be severely limited by the available system bandwidth (through I/O, network, and CPU memory hierarchy) and low data locality [10,12,24]. In large data-intensive applications like TPC-H and Map-Reduce, the dominant computations on data are scan and filtering, aggregation, sorting and join; data sets flow from the storage (through network) to all memory hierarchy levels in the host, only to be touched by a host CPU briefly. Bandwidth mismatch along the data access path results in performance loss and moving around the massive data consumes sizable energy. This unwarranted inefficiency, of both performance and energy, remains a serious roadblock to database systems to scale out.

In order to significantly improve the efficiency of processing large data sets, this paper proposes and explores the idea of accelerating database operations for data warehouse workload by moving all or portions of data-intensive processing to inside flash SSDs, close to the data source itself (i.e., flash memory chips). We refer to this approach as "*in-storage processing*" (*ISP* in short) because data processing is performed inside a storage device.

As large data-intensive applications are popular and their demands for data processing grow exponentially, the current computing paradigm of bringing data to host CPU for computation will encounter the unprecedented "bandwidth crisis" along the path from storage, network, DRAM to CPU. The contemporary solution with the simple Map-Reduce programming paradigm on massively large numbers of commodity PC clusters would be also sub-optimal because it also brings data to the host CPU. A more fundamental solution is to bring the computation close to data itself, and thus to remove the potential bandwidth bottleneck. Fortunately, owing to the advent of the bandwidth breakthrough in flash memory and the intrinsic parallelism inside an SSD, it is the right time to revisit the concept of database machines and active disks with the cost-effective System-on-Chip (SoC) technology. As we demonstrated in this paper, ISP can be a very promising solution for the next generation data-intensive computing paradigm in terms of performance, cost, and energy.

The idea of intelligent processing and data storage together was examined previously. In the late 1970s and early 1980s, this idea was actively studied in the context of database machines [5]. These systems became out of favor from the late 1980s mainly because their modest performance gain did not justify the high cost of special-purpose hardware [5]. The idea was then revisited and extended in the late 1990s to using an array of commodity active disks that integrate embedded general-purpose processors and memory [23,36].

There is a fundamental difference among our approach, the database machine and the active disk approach. The concept of active disk was mainly motivated by the superfluous computing power inside individual disks. In a similar vein, as Boral and DeWitt concluded in their retrospect paper for database machines [5], the limiting factor was the I/O bandwidth of storage media (i.e., disks), not the CPU processing power or the DRAM bandwidth. Therefore, the previous "Disk-based ISP" was I/O bound.

In contrast, the proposed ISP with SSD is no longer I/O bound. Instead, the processing power of embedded CPU and the DRAM bandwidth inside flash SSD become new bottlenecks. Let us explain why this is the case. First, the internal I/O bandwidth of flash SSDs can be easily scaled by adopting multi-way and multi-channel interleaving of NAND flash memory chips [38]. Second, new flash memory devices provide much improved data bandwidth; recent DDR 2.0 NAND interface provides 400 Mbps of pin bandwidth [41]. Therefore, assuming an SSD with 16 channels each of which channel connects to 400 Mbps 8-bit NAND flash memory, the aggregated raw bandwidth amounts to as high as 6.4 GB/s. This raw bandwidth simply surpasses the computing power of contemporary embedded CPUs and DRAM. As a consequence, SSD-based ISP would be CPU-bound rather than I/O bound. Furthermore, the CPU performance is not the only bottleneck in the SSD-based ISP. Along the data path from multiple flash chips to an embedded CPU, other bottlenecks exist such as DRAM bandwidth and DRAM–CPU cache bandwidth.

Instead of exhaustively exploring the design space of the CPU-based ISP, in this paper, we turn our attention to leveraging the intrinsic data parallelism of flash SSD and deploying cost-effective data computing modules brought close to each flash chip using the System-on-Chip (SoC) technology. This "*hardware-accelerated ISP*" approach is analogos to the process-per-track and process-per-head architectures of database machines [5]. We show that the two common operators, *scan* (or *selection*) and *join* in SQL queries can be easily and efficiently implemented. In particular, each flash channel can be augmented with hardware logic that carries out a simple selection operation. While data from flash memory are transferred, the selection operation can be applied to the data stream on the fly without any delay. As a result, only the selected records will be placed in the SSD controller's DRAM. This selection operation is performed in parallel (in all flash channels) and the amount of data to be transferred to the DRAM (and ultimately to the host) can be drastically reduced, depending on the selectivity of the given filtering predicate. Then, every component along the data path within SSD can work in a balanced way as possible.