

Special issue on data management on modern hardware

Qiong Luo · Jens Teubner

Published online: 12 December 2014
© Springer Science+Business Media New York 2014

For several decades, the roles in developing IT systems remained clearly separated. It was the responsibility of the hardware community to embrace and leverage the latest technology trends. The resulting hardware would become faster—but the interfaces it exposes to software remained basically unchanged for many years. The role of the software community was to build on stable hardware interfaces and add features to the system, riding on the wave of growing hardware speeds.

The free performance ride ended about a decade ago. For a number of reasons, intricate hardware details can no longer be hidden behind simple and stable interfaces. Rather, expectations toward increased application performance can only be met if the software is made aware of hardware intricacies and its underlying algorithms are carefully tuned to exploit hardware characteristics.

The database field is particularly hit by the changing landscape. It depends on two hardware aspects that both have undergone disruptive changes in the recent years. On one side, *computing resources* have moved from a single, central processing unit (CPU) to massively parallel compute facilities. Heterogeneous hardware—fitted with co-processors such as graphics processors (GPUs) or field-programmable gate arrays (FPGAs)—has become ubiquitous; leveraging such hardware has become a tremendous challenge. On the other side, databases vitally depend on *storage*. However, the classical hard disk drive is increasingly displaced by new persistent storage media—most prominently flash memories or non-volatile RAM. Some systems dispense with persistent media altogether and perform all processing in main memory.

Q. Luo
HKUST, Kowloon, Hong Kong
e-mail: luo@cse.ust.hk

J. Teubner (✉)
TU Dortmund University, Dortmund, Germany
e-mail: jens.teubner@cs.tu-dortmund.de

This special issue features high-quality articles that cover both of these hardware aspects.

The first article, *Parallel Outlier Detection on Uncertain Data for GPUs*, uses graphics processing units (GPUs) as an instance of a specialized co-processor to accelerate data processing. The specific task addressed in the paper is the detection of outliers on uncertain data. Outlier detection is an important component of data mining systems; uncertain data arises in numerous real-world data sets.

The contribution of this first article is twofold. First, it proposes a new outlier detection algorithm which—based on density sampling—scales significantly better than previous solutions. The proposed algorithm prepares for the second contribution, a careful implementation that leverages the particular kind of parallelism realized in GPU architectures, a kernel-based programming model with warp-level scheduling. As the paper shows, the two contributions combined can lead to substantial performance improvements, compared to alternatives that are based on CPUs alone.

The second article, *Optimized B+-Tree for Hybrid Storage Systems*, studies how the changing storage landscape affects B+-tree data structures, hence, the second aspect mentioned above. Addressing B+-trees, the contributions of the article go right into the heart of most existing database engines, which are often very deeply built on B+-tree indexes.

Specifically, the article assumes a hybrid storage architecture, composed of solid state drives (SSDs) and traditional hard disk drives (HDDs). The former technology is appealing because of its favorable (random) read access characteristics, but also incurs a high monetary cost and limited lifetime characteristics (with regard to device endurance and data retention). Combining it with traditional hard disks strikes a balance between these characteristics, as demonstrated in the article. The proposed *HybridB tree* data structure leverages the hybrid combination of both drive types and ensures a data access pattern that is friendly to the characteristics of SSD media. An experimental evaluation demonstrates that *HybridB tree* can significantly outperform state-of-the-art solutions, such as using a standard B+-tree implementation on top of an SSD/HDD hybrid.

A number of people contributed to this special issue and we owe all of them a big Thank You. Specifically, we would like to thank the editors-in-chief, Divy Agrawal and Amit P. Sheth; the reviewers for their thorough comments; as well as the numerous people that work behind the scenes to make reviewing, typesetting, and publishing an easy experience for us.