Letter from the Special Issue Editor

The layering of applications, database systems, operating system, and hardware, has never been an ideal separation. For decades, database systems have overriden operating system services in order to efficiently support transactional properties. However, this layering has enabled the independent evolution of database systems based on well-defined abstractions, at a performance cost that has remained negligible for decades.

Today, this layering is challenged. First, applications that require low latency in addition to high throughput cannot tolerate inefficiencies due to layer independance. When every micro-second counts, it makes sense to streamline the data path across layers. Second, techniques and policies devised for multicore CPUs, might have to be revised for large number of possibly diverse cores. Third, the dominance of virtualized environments and the emergence of programmable hardware requires novel abstractions.

There is a need to rethink the boundaries of applications, database systems, operating system, and hardware and revisit the nature of the interactions across these layers. In this issue, we have collected six articles that illustrate various forms of cross-layer support.

The first article proposes to share information across layers to optimize resource utilization. It is based on experimental results with SAP HANA and proposes a concrete instance of co-design between Database and Operating System. It identifies the need to bridge the semantic gap between application and database system.

The second article illustrates a mechanism, hyperupcalls, specifically designed to bridge such a semantic gap. Hyperupcalls have been designed to bridge the semantic gap between a hypervisor and a guest virtual machine (this work got the best paper award at Usenix ATC 2018). In this paper, the authors explore how hyperupcalls could be used from the perspective of applications and database systems.

Following these two articles from industrial research, the following three articles are authored by young faculty members. Each outlines an ambitious research agenda that revisits the role of database systems on the data, control, and compute planes of modern computers.

The third article revisits operating system support for data management on modern hardware. It argues for new forms of interface between database system and operating system. The fourth article focuses on specialized hardware and the opportunities this creates for database system design. The fifth article details the challenge of utilizing server cores efficiently for data intensive tasks, especially on servers equipped with many/diverse cores, and makes the case for cross-layer support for data-intensive task scheduling.

The sixth article focuses on a general purpose synchronization method, optimistic lock coupling, that is both simple and scales to a large number of cores. This method is applicable to most tree-like data structures and should be considered as an operating system-level facility on multi-core CPUs.

The traditional layering has favoured contributions from independent communities. The cross-layer approaches outlined in this issue, favour collaborations between database experts and experts on hardware, operating systems, compilers, and programming languages. We anticipate that the data engineering community will increasingly engage with these communities in the near future.

Philippe Bonnet
IT University of Copenhagen