Databases have been around for quite a while now. A great success story, today's databases are built on very solid theoretical foundations and an amazing wealth of engineering knowledge. Yet, for all their success, some of the fundamental assumptions behind the architecture of modern database systems are being challenged by the evolution of the underlying hardware. In this issue of the Bulletin, several authors explore the problem in great detail and provide valuable insights on how to exploit new features at the hardware to improve database performance.

The first article, by Jim Gray, reviews the history of a number of database benchmarks. It points out that the rate of improvement in speed for several key benchmarks has slowed down in the last years. He conjectures that the reason for this slowed improvement is due to the benchmarks becoming bound by memory latency, and the slow progress in speed merely reflecting the slow improvement in memory latency. The second article, by Ken Ross et al., discusses how different aspects of modern processors can be exploited by database engines. It provides a very good overview of the different issues involved and of a number of techniques proposed over the years to adapt database processing to the characteristics of the underlying hardware. While Ross' article describes a collection of techniques, the third article in the issue -by Harizopoulos and Ailamaki- proposes a novel database architecture that optimizes data and instruction locality at all levels of the memory hierarchy. The relevance of this contribution is that it shows what needs to be changed in the way databases are built to be able to function efficiently on modern processors. The fourth article in the issue, by Zukowsky et al., explores the problem of minimizing the amount of instructions needed to process a query. They show that the performance of today's databases is order of magnitudes worse than hand coding the solution using C++. Then they go on to describe a new query execution engine, X100, that uses in-cache vectorized processing to reach very significant performance improvements. The last article, by Bandi et al., tackle the problem of using new hardware such as MEMS storage devices to improve the performance of certain database operations. The article shows how to effectively use MEMS to process spatial queries and they show how to incorporate this new hardware into existing database products.

Taken together, these articles offer a fascinating perspective on how some of the basic assumptions behind the architecture of existing database systems are being challenged by advances in hardware. If we consider the changes that are taken place in many other areas (e.g., use of clusters for scalability, or the advent of aspect oriented programming) it becomes clear that databases must radically change their architecture to avoid becoming niche systems usable only in a very small set of applications. I am sure some of these topics will soon become the main theme of new issues of the Data Engineering Bulletin.