

Letter from the Editor-in-Chief

More IEEE Computer Society Activities

I have taken a new role as co-chair of the Conferences Advisory Committee within the IEEE Computer Society. This led me to attend the Society's meeting in Long Beach, CA in February, and exposed me to the on-going efforts of the Society to adapt to changing circumstances.

Organizational time, of necessity, moves slowly compared with human time scales. Changing the Computer Society is, in some sense, like changing widely used software. Bad things happen if change is not done carefully. So the only news I have to report is that the people in the Computer Society, volunteers and staff, understand that change is required and a number of options are being considered.

I think there are three primary facets to the current Computer Society situation. (1) Financial: the decline in publication revenue requires rethinking the Computer Society "business model". (2) New technology: Mobility, cloud, social networks, and online activities in general create opportunities for an expanded role. (3) Technical Reputation: Any changes must preserve the reputation of the Computer Society as a "brand" synonymous with technical quality. That limits what can be done but enhances the value of the things that are done.

So the Computer Society is actively seeking a way forward. If you have thoughts on this, I would be very interested in hearing them. Working together is amazingly effective in such an endeavor.

The Current Issue

How languages "play" with database systems has always been an important issue. We (the database technical community) optimize queries, and then execute the queries using a database management system. This has not changed. We build database systems using languages. This has not changed. But.... in a very real sense, almost everything has changed.

One change is how much we want substantially higher performance for our queries and updates. This quest for performance has led to major re-architecting of database systems. Elements designed decades ago have been re-designed to work well on modern multi-core processors, exploiting enormous main memories, and sometimes using flash disks (SSDs). As performance of the system elements has increased, the fraction of query execution time that is actually consumed by the query specification/program using the database system has increased dramatically. This has led the database community to increase their exploitation of compiler technology.

Compiler generation of highly efficient query code is one aspect of the picture. Looking at the division of work between database system and application suggests that the boundary has some interesting flexibility that compiler technology might help us exploit. These are just two of the several threads to the work reported here. Sudarshan has assembled a cross-section of the work in this area, from query code generation to database system implementation, etc. This area has had a growth spurt that makes it both a very old and a very young area- characteristic of an area in turmoil, and primed to change the nature of our systems. This is a subject that really demands careful study and this issue is a great place to start.

David Lomet
Microsoft Corporation