CPUs and GPUs: Who Owns the Future?

ERIK R. ALTMAN
caltman@us.ibm.com

The CPU versus GPU debate seems to arouse great interest and passion—more so than most any other topic in microarchitecture right now. One bit of evidence for this assertion comes from the fact that we received twice the typical number of articles for this special theme issue, an interest reflected in many other publications and venues as well. Perhaps this intense interest stems from the high academic and economic stakes and the uncertainty of the outcome.

Perhaps it is also that GPUs are attempting something audacious. I assert that successful chip styles all fall on a line in which Performance / (Programming time) is a constant, as illustrated in Figure 1. As technology improves, each style of chip improves by the same proportion—that is, the relative ease of programming does not change. GPUs with attendant languages such as CUDA and OpenCL are trying not only to improve performance but to reduce programming time, as illustrated in Figure 2, thus encroaching on traditional CPU turf.

Although the final trajectory for GPUs is unclear, GPU success would raise the question of whether GPUs are unique or whether other chip styles such as field-programmable gate arrays (FPGAs) and application-specific integrated circuits (ASICs) could also move toward the left in Figures 1 and 2.

However, it is not clear whether it is essential that GPUs ultimately reach the CPU point in Figures 1 and 2. Increased
programmability and performance of GPUs will increase the amount of software written for them, and if total person-years spent writing software is fixed, increased GPU activity will of necessity reduce CPU software activity. This switching trend would likely be accelerated if GPUs opened up a new killer app. Although a number of workloads gain large benefit from GPUs, a killer app has not yet emerged.

However, CPUs are not standing still. With things like wider single instruction, multiple data (SIMD), CPUs have also been trying to move above the constant lines in Figures 1 and 2. Efforts on FPGAs and ASICs are likewise not standing still. Given all of these efforts, the result could be that everything stays on the same line, but the line moves up more quickly. After all, this is what a constant value of Performance / (Programming time) predicts.

The excellent set of articles in this issue effectively addresses these and many more topics, and I thank David Brooks for serving as Guest Editor. Not only did David recruit and drive the review and selection process for this outstanding set of articles, but he dealt with the large number of submissions efficiently.

We are always on the lookout for other topics of keen interest to the community. If you have suggestions, please e-mail me at ealtman@us.ibm.com.

Erik R. Altman
Editor in Chief
IEEE Micro

Selected CS articles and columns are also available for free at http://ComputingNow.computer.org.