POINT OF VIEW

TEXAS INSTRUMENTS

By
William Mills,
Distinguished Member of Technical Staff

A new look at applications and multiprocessor architectures

About the author

Bill Mills is Chief Technologist for Open Linux Solutions in TI's Software Development Organization and focuses on non-wireless handset applications. Before taking the current position, Bill was the primary microprocessor software architect for TI's voice-over-packet product and the system architect for TI's high-density voice platforms and products. He joined TI in 1995 and has been programming professionally for 22 years.

The world of 2020 promises to be a very different place than the one we live in today. Certainly, what we consider "commodity" processing today would have surprised anyone 12 years ago – for example, a $7 optical mouse has a CCD and 7 MIPS DSP under the hood – and so we must assume the future shall surprise us in a similar way over the next 12 years.

While it will be difficult to predict what will become a commodity technology, both fixed-function and general-purpose architectures will continue to play significant roles going forward. Fixed function will be the favorite for high-volume commodity items, but general-purpose architectures will still have the advantage of spreading technology development costs across a wider range of lower-volume applications. What will change is where the balance point lies for each application.

One predominant design concern is data flow. In the past, processor architectures have focused on optimizing instruction and operation execution. However, since processing speed increases at a faster rate than memory speed, chip designers will not be able to place enough memory close to the processing elements. In order to overcome the resulting throughput bottleneck, there will need to be increased data sharing between multiple processing elements. Put another way, as the number of processing elements increases, multiprocessor programming begins to become more like network application programming.

For this reason, TI is turning to the web to understand how to efficiently manage data flow. A network's infrastructure is built from a combination of elements, some general purpose in nature, to handle generic tasks like web and e-mail transactions while others are more specialized, such as NAS (Network Attached Storage) components. When capacity is important, soft partitioning dynamically shares the processing load between elements. When predictability is required (i.e., real-time or latency-sensitive functions), hard partitioning assigns different work types or scopes to fixed clusters.

The more fine-grained processing elements become, the more impact OS threads and operations will have on system performance. For example, if threads are to be suspended during data requests and potentially resumed on another core, context switching must be minimized. To prevent OS bottlenecks, micro kernels will come back into vogue and many of their functions – timer call backs, mutexes, queues and lists, context save and restore, task scheduling and message passing, to name a few – will migrate to hardware.

Carrying these concepts to processor architectures will result in an asynchronous and transaction-based development model. For example, function calls will be treated as distributed processing requests that can be spread over multiple-processing elements transparently to the developer and to the application. In order to avoid blocking of sequential operations, asynchronous or event-based processing will become the norm. Transactions will organize and provide a higher level structure to event-based programming. In addition, "restart-able" transactions will increase reliability.

While this vision is admittedly progressive – procedural programming has proven quite resistive to paradigm shifts over the last 30 years – there is precedent today in networking frameworks (i.e., Python's "Twisted" Framework). In any case, programming multiprocessor architectures requires a new way of looking at applications. We'll see how right I am over the next 12 years.