Forward ... A Foreword

Professor Bruce Jacob
University of Maryland

Using present-day architectural design concepts to drive forward the design of next-generation large-scale systems is an attractive approach but is inherently misguided; today's methods simply do not scale to tomorrow's sizes, and no amount of forcing will cram that square peg into a round hole. Nowhere is the mismatch more apparent than in the memory system, because the memory system constitutes a majority of the silicon in a large-scale system. The following points illustrate just a few of the engineering challenges facing us:

- *Power.* Power per node in large-scale systems is on the order of 100 W, a conservative number, roughly half of which is dissipated in the memory system (DIMMs dissipate roughly 20 W when in use, which means 20 W per channel; today's systems often have three or more DRAM channels per CPU socket). Future installations are expected to have hundreds of thousands of nodes. Multiplying the two yields power requirements in the 10–100 MW range per installation, roughly half of which will be dissipated by memory. With electricity costing USD 1 million per megawatt-year, following the current design trend will cost tens of millions of dollars per year per installation, just for the electricity to compute, never mind the cost of cooling.
- *Reliability.* Memory chips comprise the largest number of chips in a typical system. For instance, in a typical drawer one will find on the order of 10 CPU chips, 100 supporting chips (I/O, administrative and monitoring, and other glue), and 1000 DRAM chips. For a medium- to large-scale system (10–100 racks of 10–100 drawers each), this would imply something on the order of 1 million DRAM chips system-wide. Reliability in such a system becomes a significant issue: even with DRAM hard-error FIT rates better than 10, this yields a statistically guaranteed hard-error device failure somewhere in the memory system every few days. Soft errors are generally one to two orders of magnitude more frequent than hard errors, meaning that transient errors will occur at the rate of once every few hours.
- *Volume.* One of the more significant costs of a computing installation is the physical plant, which scales with the physical volume of the computing circuitry needed (that is, how much space it takes up). Simply put: more computing performance requires more computing circuitry, which requires a larger building. Figure out a way to reduce the volume of the circuitry required—in particular, the volume of the memory system required (see previous point)—and you can reduce the size of the physical plant needed. Note that there are a handful of obvious ways to reduce the volume of the memory system, including reducing the total number of bits (not particularly appealing while the number of cores is increasing), reducing feature size (the present approach), or changing to another memory technology with significantly different density characteristics. Note also that, besides reducing the cost of construction, reducing the physical size can also reduce the cost of cooling the computing circuitry (for example, by reducing the number of chillers and air handlers needed), a cost that typically represents half the overall power budget.

It should be clear from this brief look at the challenges facing us that moving forward in large-scale system design will require significant work at the memory-system level. Because the memory system imposes such significant limitations (including performance, power, reliability, and space), we cannot move forward without understanding these limitations and fixing them, which is likely to require a redesign of memory systems in general.

These are challenges of *efficiency and reliability*. One way to look at large-scale installations (supercomputers, and most enterprise-computing systems as well) is that they are the world's highest-performance embedded systems. Most embedded systems are only valuable if they are *efficient* (for example, when they run on a battery charge all day long) and *reliable* (work correctly and require little system maintenance). Like embedded systems and unlike typical general-purpose systems, large-scale installations tend to run the same software 24x7. Like embedded systems and unlike typical general-purpose systems, users of these installations will go to great lengths to optimize their software and often write their own operating systems for the hardware. And, most importantly considering the focus of the special issue you are reading, like embedded systems, efficiency and reliability in large-scale systems is now (or now has become) the key point. In the design of tomorrow's large-scale systems, people care more about efficient and reliable solutions than high-performance solutions—not because they *want* to, mind you, but because they *have* to. Whereas in the past, performance or capacity sometimes came at a high price in power or reliability, today that is no longer an acceptable tradeoff. The best solutions for tomorrow's systems will be the ones that promise *reliability and efficiency*, even if at a modest cost in performance or capacity.

The articles in this special issue of ITJ address precisely these problems and from precisely this perspective. The articles in the *Low Power Cache/Memory* section trade off cache/memory capacity for reliability and lower power. The articles in the *Error-Correcting Codes* section provide advanced reliability techniques wherein reliability is ensured through redundancy. The *Invited Academic* articles address reliability and lifetime concerns of flash memory. Last, the *Evaluation and Infrastructure* article describes a new simulation framework for accurately evaluating memory-system designs that integrate nonvolatile technologies directly into the memory hierarchy.