FORUM

Why Computing Parallels Brain Function

Peter J. Denning's "The Profession of IT" column ("The Locality Principle," July 2005) invoked an anthropomorphic explanation for the prevalence of the locality principle in computational systems, observing that humans gather the most useful objects close around them to minimize the time and work required for their use, and that we've transferred these behaviors into the computational systems we design.

A more intellectually satisfying explanation might be that we are dealing with two parallel and independent evolutionary design paths. Trading some expensive high-quality space (fast memory) in order to gain time performance is a sound engineering decision. It is therefore likely that evolution first adapted the human brain by endowing it with limited but versatile short-term memory and large long-term memory—a structure that exhibits behavior similar to caching.

Millennia later, we make similar design decisions when building computing systems.

Diomidis Spinellis
Athens, Greece

Author Responds

I agree completely. The human brain's structure makes locality a natural feature of human behavior. Spinellis offers an explanation grounded in evolution. There are undoubtedly subtle points. For example, when short-term memory is full, the decision about which item is to be deleted next probably depends on context and not just on which item hasn't been used in a while.

Peter J. Denning
Monterey, CA

I take issue with some of Peter J. Denning's conclusions (July 2005). The "locality" principle he discussed is really Zipf's law in disguise, and many of the milestones he cited (such as Akamai's Web caches) are merely examples of Zipf's law in action. Further, Zipf provided a mathematical formula that could be tested, while locality is a qualitative observation.

Most operating systems are not particularly good at paging, but cheaper and cheaper main memory has minimized the pain of paging. Indeed, one can now routinely put more main memory on a PC than its processor is able to address.

Finally, Denning made quantitative claims about the ability of virtual memory to improve "programmer productivity." While I am a fan of the programming model of virtual memory and the simplifications it allows the programmer, I don't recall any studies of "programmer productivity" that would support this claim. Indeed, the optimizations required to deal with limited main memory don't go away with virtual memory but take on a slightly different guise. The programmer's productivity for data-intensive main-memory-limited problems (such as sorting and database operations) is only mildly improved through virtual memory. (Seymour Cray was right.)

Henry Baker
Encino, CA

Author Responds

Baker is apparently unhappy because he thinks computer scientists are trying to take credit for a concept already understood from Zipf's law and because virtual memory

PAUL WATSON