DOI:10.1145/2330667.2330669

Operationalizing Privacy by Design

While I largely agree with Sarah Spiekermann's Viewpoint "The Challenges of Privacy by Design" (July 2012), as Ontario's Information and Privacy Commissioner (http://www.ipc.on.ca/english/Home-Page/) and originator of Privacy by Design (PbD http://privacybydesign.ca/), I want to make it clear that many organizations do understand the need to embed privacy in their systems and technologies and regularly ask my office's help in doing so.

I called 2011 "The Year of the Engineer" because of all the companies that had asked me to speak to their engineers and software designers about how to embed privacy, by design, into their operations. Ranging from major players like Adobe, Google, and RIM to smaller ones, all got the message that embedding privacy from the start is critical to their operations. Indeed, my message seemed well received and welcome.

The next stage of PbD evolution is to translate its "7 Foundational Principles" (http://privacybydesign.ca/about/principles/) into more prescriptive requirements, specifications, standards, best practices, and operational-performance criteria. To provide guidance, we are presently writing a how-to paper on operationalizing PbD that should provide further implementation assistance in specific areas. It will supplement earlier papers like "Operationalizing Privacy by Design: The Ontario Smart Grid Case Study" (http://www.privacybydesign.ca/content/uploads/2011/02/pbd-ont-smartgrid-casestudy.pdf), a collaborative effort by my office, GE, Hydro One, IBM, and Telvent published in 2011 to demonstrate how PbD was being operationalized in a major smart-grid project.

Following and sharing the principles with technologists and senior management puts an organization well on its way to strong privacy. Translating them into concrete steps is not difficult yet ensures privacy is embedded into an organization's operations, by design. We have shown how it works with our partners in North America and Europe by embedding PbD into smart-meter and smart-grid designs. Another example is the Ontario Lottery and Gaming Corporation, which operationalized PbD in its face-recognition system for casinos and gaming facilities to identify participants of a voluntary self-exclusion program for problem gamblers while fully protecting the privacy of all other patrons (http://www.ipc.on.ca/english/Resources/Discussion-Papers/Discussion-Papers-Summary/?id=1000).

The challenges of PbD are not as great as Spiekermann suggested; the engineers I have met have embraced the PbD principles, finding implementation not difficult. Perhaps all that is needed is to put PbD on engineers' radar screens and empower them to develop and adopt privacy best practices, share their implementation experiences, and provide recognition for innovative solutions.

Ann Cavoukian, Ontario, Canada

Local (and Coherent) Caches Not Such Obvious Choices for Multicores

To appreciate why a key assumption of "Why On-Chip Cache Coherence Is Here to Stay" by Milo M.K. Martin et al. (July 2012)—that on-chip multicore architectures mandate local caches—may be problematic, consider the following examples of a shared variable in a parallel program a processor would write into:

Example 1. No other processor seeks to read or write the variable, in which case little harm is done copying the variable to a location local to the processor (register or scratch pad), accessing it as needed, and then storing it back to a shared location or variable; and

Example 2. Other processors need to read from and/or write into that variable.

The two local-cache cases compared in the article, with or without cache coherence, require considerable traffic to ensure coherent access to the variable. However, if all write updates in a parallel program are done to a shared location using prefix-sum or other transactional-memory type instruction, traffic is proportional to the actual number of accesses to that variable by all processors. This way of performing write updates represents a significant improvement over the automatic cache coherence advocated by Martin et al. in which every access requires broader notification.

Overall, the bet Martin et al. advanced in the article on large private caches for parallel on-chip computing has yet to prove itself as a good allocation of silicon resources; for example, in a 1,000-core design, one more word in each private cache could mean 1,000 fewer words in shared cache—not necessarily a good deal in terms of ease of programming and overall performance.

My own recent research[1,2] at the University of Maryland suggests the traditional emphasis on private caches could be the main reason programming current multicores is still too difficult for most programmers. Though the code-backward-compatibility argument is compelling for serial code, the difficulty of parallel programming in general, and for locality in particular, remains the biggest obstacle inhibiting adoption of multicores.

Uzi Vishkin, College Park, MD

References

1. Vishkin, U. *Computer Memory Architecture Methods for Hybrid Serial and Parallel Computers*; U.S. Patents 7,707,388 and 8,145,879.
2. Vishkin, U. Using simple abstraction to reinvent computing for parallelism. *Commun. ACM 54*, 1 (Jan. 2011), 75–85.

Communications welcomes your opinion. To submit a Letter to the Editor, please limit yourself to 500 words or less, and send to letters@cacm.acm.org.