Active-Page Programming: New Paradigms for Using Intelligent Memory

Frederic T. Chong, Mark Oskin, Justin Hensley, Aneet Chopra, and Lucian Vlad-Lita
Department of Computer Science
University of California at Davis

Modern machines suffer from a growing processor-memory performance gap. Intelligent memory systems can address this gap by moving computation from the processor into the memory system [P^+97] [KADP97]. We have proposed *Active Pages*, a page-based model of intelligent memory designed to support commodity microprocessors [OCS98]. Active-Page memory systems augment the classical von Neumann architecture by providing simple computations associated with each page of data. Central to this model is the *partitioning* of applications between processor and memory. This partitioning leads to entirely new paradigms for programming machines of the future.

An Active Page consists of a superpage of data and a set of functions that operate on that data. For example, the data may be an array and the functions may be insert, delete, and find elements. Active-Page systems exploit DRAM technology to provide inexpensive computation which scales with dataset size. Active-Page implementations generally consist of computational logic coupled with each sub-array of memory on a DRAM chip. These additional computational resources lead to programming tradeoffs which are substantially different from conventional systems.

For example, programs on conventional systems must often choose between datastructure flexibility and efficiency. A linked list can be easily modified, but an array is more efficiently accessed. Active Pages, however, allow data to be stored as arrays, yet quickly modified through in-memory copying.

Similarly, sparse datastructures are often chosen to save memory, but lead to poor cache performance. Active Pages can perform in-memory gather-scatter to present the processor and cache with dense, regular data even though the underlying data is sparse or irregular.

Furthermore, complex algorithms with lower computational growth but high algorithmic complexity are chosen over computationally challenging but conceptually simple algorithms. Cheap, ubiquitous parallelism alters the costs in terms not only of program maintainability, but also potentially of overall run time of algorithmic implementation. What is simplier, but slower in the classical von Neumann architecture may be faster and more maintainable in the Active-Page architecture.

Finally, automatic memory management is often thought to have too large of a negative impact on performance. In-memory incremental garbage collection could avoid such concerns yet provide the convenience and safety of automatic memory management.

Active Pages alters the fundamental programming model of computing devices, while leaving in place the bulk of the existing hardware mechanisms and structure. Clearly, to take full advantage of the computational potential, new software tools, abstractions, algorithms and paradigms need to be developed. These new tools may leverage parallelizing compiler technology [HAA^+96] for in-memory computation and hardware-software co-design methodology [VG95] for computation partitioning.

Intelligent memory systems are coming. The programmer or compiler of the future must divide work between processor and memory system. Data-intensive tasks which were computationally expensive in conventional systems will become commonplace in the future. In-memory computation will free the user to make a wider variety of programming choices for both convenience and performance.

References

[HAA^+96] M. W. Hall, J. M. Anderson, S. P. Amarasinghe, B. R. Murphy, S.-W. Liao, E. Bugnion, and M. S. Lam. Maximizing multiprocessor performance with the suif compiler. *Computer*, December 1996.

[KADP97] Kimberly Keeton, Remzi Arpaci-Dusseau, and David A. Patterson. IRAM and SmartSIMM: Overcoming the I/O bus bottleneck,. In *Workshop on Mixing Logic and DRAM: Chips that Compute and Remember*, Denver, Colorado, June 1997.

[OCS98] Mark Oskin, Frederic T. Chong, and Timothy Sherwood. Active pages: A computation model for intelligent memory. In *Proceedings of the 25th Annual International Symposium on Computer Architecture (ISCA'98)*, Barcelona, Spain, 1998. To Appear.

[P^+97] D. Patterson et al. The case for intelligent RAM: IRAM. *IEEE Micro*, April 1997.

[VG95] F. Vahid and D. Gajski. SLIF: A specification-level intermediate format for system design. In *Proceedings of the European Design and Test Conference*, pages 185–189, Washington, March 6–9 1995. IEEE Computer Society Press.