The Hamal Parallel Computer

J.P. Grossman

Artificial Intelligence Laboratory
Massachusetts Institute of Technology
Cambridge, Massachusetts 02139

http://www.ai.mit.edu

The Problem: Over the years there has been an enormous amount of hardware research in parallel computation. It is a testament to the difficulty of the problem that despite the large number of wildly varying architectures which have been designed and evaluated, there are few agreed-upon techniques for constructing a good machine. Even basic questions such as whether or not remote data should be cached are unanswered. This is in marked contrast to the situation in the scalar world, where many well-known hardware mechanisms are consistently used to improve performance (e.g. caches, branch prediction, speculative execution, out of order execution, superscalar issue, register renaming, etc.).

Motivation: The primary reason that designing a parallel architecture is so difficult is that the parameters which define a ”good” machine are extremely application-dependent. A simple physical simulation is ideal for a SIMD machine with a high processor to memory ratio and a fast 3D grid network, but will make poor utilization of silicon resources in a Beowulf cluster and will suffer due to increased communication latencies and reduced bandwidth. Conversely, a parallel database application will perform extremely well on the latter machine but will probably not even run on the former. Thus, it is important for the designer of a parallel machine to identify the target application space in advance

There is an obvious tradeoff involved in choosing an application space. The smaller the space, the easier it is to match the hardware resources to those required by user programs, resulting in faster and more efficient program execution. On the other hand, machines with a restricted set of supported applications are less useful and not as interesting to end users. As a result, they are not cost effective because they are unlikely to be produced in volume. Since not everyone has $100 million to spend on an architecture, there is a need for commodity general-purpose parallel machines.

Approach: The overall goal of the Hamal project is to investigate design priniciples for general purpose parallel architectures. Our focus is on three specific issues:

Silicon Efficiency: In a parallel machine with many processors on each die, overall silicon efficiency (roughly defined as performance per unit area) is more important than the raw speed of any individual processor. To improve silicon efficiency, the Hamal architecture specifies a multithreaded VLIW processor with in-order execution and hardware predication.

Massive Scalability: There are two problems which are not addressed in a scalable manner by existing memory systems. The first of these is locating a piece of data within the machine given its virtual address. Typically this information is folded into Translation Lookaside Buffers (TLB's). The TLB's are then essentially a cache for a large data structure which must be kept globally consistent, and as such suffer from the same scaling difficulties as coherent data caches. The second problem is distributed object alloction. For many parallel algorithms it is important to be able to allocate single objects in memory which are distributed across multiple nodes in the system. The challenge is to allow arbitrary single nodes to perform such allocations without any global communication or synchronization.

RAM Integration: Over the past few years, several manufacturers have started offering processes that allow CMOS logic and DRAM to be placed on the same die. In its simplest form, this technique can be used to provide existing processor architectures with low-latency high-bandwidth memory [10]. A more exciting approach is to augment DRAM with small amounts of logic to extend its capabilities and/or perform simple computation directly at the memory. While Several research projects have investigated various ways in which this can be done (e.g. [7, 6, 5, 3]), none of the proposed architectures are general-purpose due to restrictions placed on the application space and/or the need to associate a significant amount of application-specific state with large portions of physical memory. We with to investigate embedded DRAM augmentations which support true general-purpose computing.

Previous Work: Most of the previous work related to silicon efficiency has focused on reducing complexity without sacrificing performance rather than explicitly maximizing efficiency. One mechanism which has received a great deal of attention is dynacmic out-of-order execution. In [9] it is shown that the expensive issue window can be replaced with a small number of simpler FIFO buffers without seriously impacting performance. A mechanism is proposed in [4] for achieving out-of-order execution with in-order issue logic by allowing the compiler to specify explicit delays for instructions.

Multithreading is a very well known technique. In [1] and [11] it is shown that hardware multithreading can significantly improve processor utilization. A large number of designs have been proposed and / or implemented which incorporate hardware multithreading with as many as 128 threads per processor [2].

The easiest way to integrate logic and memory is to add some basic data processing capabilities to memory and expose these capabilities to a host processor in a SIMD manner so that a restricted set of applications may be accelerated. In [3] the Terasys prototype is described which adds a 1-bit ALU to each column of memory. Active Pages support a broader spectrum of computation by associating a 256 logic element reconfigurable array [7] or a simple VLIW processor [8] with each 512KB page of data memory. Some strides towards RAM integration for general-purpose computing are made in [5], in which a highly reconfigurable processor-memory architecture is described.

Impact: The first main contribution of the Hamal project will be the presentation of novel memory system features to support a scalable, efficient parallel system. We will show how to address the problems of data location, distributed object allocation, synchronization and fowarding pointer aliasing. Central to the memory system will be a capability format which supports pointer arithmetic and nearly-tight object bounds. The second main contribution will be the evaluation of certain existing architectural mechanisms with respect to silicon efficiency. Mechanisms of interest include VLIW, hardware multithreading, and hardware page tables. The third and final major contribution will be the complete description and evaluation of a general-purpose embedded-memory parallel computer. This will provide a design point against which other general-purpose architectures can be compared.

Research Support: This work is supported by DARPA / AFOSR contract number F306029810172.

References:

[1] Anant Agarwal. Performance tradeoffs in multithreaded processors. *IEEE Transactions on Parallel and Distributed Systems, Vol. 3, No. 5,* pages 525–539, September 1992.

[2] Robert Alverson, David Callahan, Daniel Cummings, Brian Koblenz, Allan Porterfield, and Burton Smith. The tera computer system. *Proc. 1990 International Conference on Supercomputing,* pages 1–6, 1990.

[3] Maya Gokhale, Bill Holmes, and Ken Iobst. Processing in memory: The terasys massively parallel PIM array. *IEEE Computer*, pages 23–31, April 1995.

[4] J.P. Grossman. Cheap out-of-order execution using delayed issue. *Proc. ICCD '00,* pages 549–551, 2000.

[5] Ken Ma, Tim Paaske, Nuwan Jayasena, Ron Ho, William J. Dally, and Mark Horowitz. Smart memories: A modular reconfigurable architecture. *Proc. ISCA '00,* pages 161–171, 2000.

[6] Norman Margolus. An embedded DRAM architecture for large-scale spatial-lattice computations. *Proc. ISCA '00,* pages 149–160, 2000.

[7] Mark Oskin, Frederic T. Chong, and Timothy Sherwood. Active pages: A computation model for intelligent memory. *Proc. ISCA '98,* pages 192–203, 1998.

[8] Mark Oskin, Justin Jensley, Diana Keen, Frederic T. Chong, Matthew Farrens, and Aneet Chopra. Exploiting ILP in page-based intelligent memory. *Proc. MICRO '99,* pages 202–208, 1999.

[9] Subbarao Palacharla, Norman P. Jouppi, and J. E. Smith and. Complexity-effective superscalar processors. *Proc. ISCA '97,* pages 206–218, 1997.

[10] David Patterson, Thomas Anderson, Neal Cardwell, Richard Fromm, Kimberly Keeton, Christoforos Kozyrakis, Randi Thomas, and Katherine Yelick. The case for intelligent RAM: IRAM. *IEEE Micro, Vol. 17, No. 2,* pages 33–44, March / April 1997.

[11] Radhika Thekkath and Susan J. Eggers. The effectiveness of multiple hardware contexts. *Proc. ASPLOS VI*, pages 328–337, 1994.