Versatile Tiled-Processor Architectures: The Raw Approach

Rodric M. Rabbah, Ian Bratt, Krste Asanovic, and Anant Agarwal
Computer Science and Artificial Intelligence Laboratory
Massachusetts Institute of Technology
{rabbah, bratt, asanovic, agarwal}@csail.mit.edu

Advances in VLSI technology have spurred an increasing interest within the computer architecture community to build a new kind of "all-purpose" processor that is able to run a broad class of applications including primarily those from the domain of embedded systems—graphics, wireless processing, networking, and various forms of signal processing. The interest in new architectures is arguably due to the opportunity created by exponentially increasing chip-level resources, combined with the physical limits of power and wire-delay faced by today's high-performance processors. The realities of interconnect delay and power consumption seriously challenge the ability of microprocessor designers to fulfill the promise of Moore's Law, and leading microprocessor companies (e.g., Intel) are revamping their processor road maps because modern monolithic chips have reached their performance limit for the amount of power they consume. In order to overcome these physical barriers, it is necessary to rethink the conventional approach to microprocessor design, and to focus on scalable and distributed alternatives to current centralized microprocessors.

Several projects such as VIRAM [2] at Berkeley, Smart Memories [4] at Stanford, TRIPS [5] at UT-Austin, Raw [8] and SCALE [3] at MIT, and industrial efforts such as the Tarantula [1] extension to Alpha, have proposed scalable *tiled-processor architectures* that organize resources more effectively by dividing the silicon into an array of identical and programmable tiles that are connected by on-chip networks. The DARPA program in Polymorphic Computing Architectures is also a research thrust in this new area, and emerging "polymorphic" architectures will eventually compete with traditional desktop processors (e.g., Pentium 4) not so much in better performance on desktop workloads, but in *versatility*, or the ability to run a broader class of applications more effectively. We also expect that architectures that are more versatile are also likely to run complex real-world applications more effectively, since complex applications are often comprised of diverse components.

An example of a versatile tiled-processor architecture is the Raw microprocessor which was designed and implemented at MIT. Raw divides the chip into a two-dimensional mesh of sixteen programmable tiles, and interconnects them through an on-chip, point-to-point scalar operand networks (SON) [7]. The Raw processor can issue sixteen different floating-point, integer, load, store, or branch instructions each cycle. It also has a large set of registers and a distributed memory hierarchy. The SON is exposed to the Raw compilation infrastructure which orchestrates the flow of data within the network for streaming computation and fine-grained instruction-level parallel-processing.

The focus on new kinds of architectures and architectural versatility necessitates *new benchmark suites and metrics* to accurately reflect the goals of the architecture community. Toward that end, we propose *VersaBench* as a new benchmark suite, and *Versatility* as a new metric. VersaBench is a collection of applications from three market-dominant areas: desktop, server, and embedded computing. In the desktop class, we distinguish between integer benchmarks and floating-point benchmarks (which are synonymous with scientific benchmarks). We view the server class in a broad throughput-biased perspective, spanning transaction-processing, web-services, and grid-computing (e.g., ergonomics and material science industrial research). The embedded category is characterized by streaming and bit-level computing. The VersaBench constituents thereby serve to adequately reflect the broad set of workloads that new architectures are required to run. The suite is available online at `http://cag.csail.mit.edu/versabench`.

The Versatility of an architecture is the geometric mean of the speedup of every application in the VersaBench suite relative to the architecture that provides the *best* performance for that application (in the 2004 time frame from known results at the time of this writing). The Versatility may be separately normalized by chip area, power or machine cost. This new metric is inspired by SPEC rates [6]. For example, the SPEC CINT89 rate for an architecture is the geometric mean of the speedups of that architecture relative to a reference machine (e.g., the VAX 11/780) for each of the applications in the SPEC CINT89 suite. Note that because Versatility normalizes performance relative to the best processor for each application, it not just another geometric mean over N benchmarks. The Versatility measure tells us whether there is opportunity to improve an architecture, and where the effort should be spent. For example, if the performance on streaming benchmarks is not up to par, then supporting a streaming data-memory is a better choice to increasing the size of the instruction cache.

Table 1. Characteristics of the VersaBench workloads.

benchmark category	data type	parallelism	control complexity	temporal locality	spatial locality
Desktop Integer	integer	low	high	high	low
Desktop Floating-Point	float	medium	medium	medium	medium
Server	integer/float	high	medium to high	medium to high	medium to low
Embedded Streaming	integer/float/bit	very high	low	low to high	very high
Embedded Bit	bit	very high	very low	very low	very high

Presentation Outline: The presentation will describe the Raw architecture, its implementation, and performance. We will focus on Raw's ability to support (i) the diverse set of applications embodied by the VersaBench suite and (ii) multiple forms of parallelism, including instruction-level-parallelism (ILP) for desktop applications, and stream parallelism for embedded computing. We will also report detailed performance measurements that quantify the versatility of Raw compared to some widely deployed architectures. As a prelude, the measured versatility of the Raw processor is 0.7, while that of the Pentium III is 0.1. The Pentium's relatively poor performance on stream benchmarks hurts its versatility, and although Raw's versatility is better in comparison, the VersaBench suite highlights two clear areas that merit additional research. The first is in improving the architecture to better support embedded bit-level workloads: ASICs perform $2-3\times$ better than Raw. Another area of research focuses on desktop integer applications: Raw's performance is $2\times$ lower than a Pentium III for applications with low degrees of ILP.

VersaBench: The VersaBench suite consists of fifteen benchmarks: three benchmarks in each of the five categories that makeup the desktop, server, and embedded application workloads. The VersaBench applications were selected systematically from a pool of candidates that exceeded the target number of benchmarks in the suite. For each candidate application, the selection process focused on measuring the following *basis* properties of the program:
— *predominant data type*: summarizes the predominant type-domain over which computation is performed,
— *parallelism*: quantifies maximum IPC (instructions per cycle) in a benchmark,
— *control complexity*: measures instruction temporal locality,
— *data temporal locality* and *data spatial locality*
Intuitively, we believe the basis properties of the five benchmark-categories are as shown in Table 1. Accordingly, the VersaBench suite was created by measuring the properties of several applications and selecting those that match intuition. The presentation will include results that map the VersaBench constituents in the space of basis properties.

Versatility Metric: The Versatility of an architecture is defined as the geometric mean of the speedup of every application in the VersaBench suite relative to the architecture that provides the best performance for that application. We use the geometric mean because it has a damping property that is desirable when measuring versatility: *it is harder to bias the versatility measure of an architecture simply because the architecture performs extremely well on a single application.* This is because the mean will increase proportional to the N^{th} root of the speedup, and therefore, one application cannot skew the results significantly. The metric is designed to quantify the versatility of an architecture. For example, ASICs or application specific integrated circuits have a Versatility of zero since they are highly specialized. Also note that as future process-technologies deliver higher clock frequencies, architectural versatility will increase.

The Versatility metric is used in one of two ways when considering the evolution of processors and their performance. In one way, we select the best architecture for each application at the time of this writing (2004), and we always use their respective running times to normalize speedups. This approach provides a common standard for all time but has the drawback that as machines get faster over time, their Versatility eventually surpasses unity. This is not counter intuitive however, since faster processors can run more applications effectively. The SPEC analogy is to normalize to the performance of a VAX 11/780 for all time. In an alternate approach, we can renormalize to a new set of "best" machines every few years, so that the Versatility of processors is always below unity[1]. The process of renormalizing for older machines is easy and does not require a knowledge of the individual application speedups.

[1]The reference machines for SPEC have changed over time. While the VAX 11/780 was the reference machine for SPEC CINT89 and SPEC CINT92, the SPARCstation 10/40 was the reference machine for SPEC CINT95, and the Sun Ultra5-10 workstation with a 300MHz SPARC processor is the reference machine for SPEC CINT2000.

REFERENCES

[1] R. Espasa, F. Ardanaz, J. Gago, R. Gramunt, I. Hernandez, T. Juan, J. Emer, S. Felix, G. Lowney, M. Mattina, and A. Seznec". Tarantula: A Vector Extension to the Alpha Architecture. In *Proceedings of the 29th International Symposium on Computer Architecture (ISCA)*, pages 281–292, 2002.

[2] C. E. Kozyrakis and D. Patterson. A New Direction for Computer Architecture Research. *IEEE Computer*, 30(9), Sept. 1997.

[3] R. Krashinsky, C. Batten, M. Hampton, S. Gerding, B. Pharris, J. Casper, , and K. Asanovic. The vector-thread architecture. In *Proceedings of the 31st International Symposium on Computer Architecture (ISCA)*, 2004.

[4] K. Mai, T. Paaske, N. Jayasena, R. Ho, W. Dally, and M. Horowitz. Smart memories: A modular reconfigurable architecture. In *Proceedings of the 27th International Symposium on Computer Architecture*, pages 161–170, 2000.

[5] R. Nagarajan, K. Sankaralingam, D. Burger, and S. W. Keckler. A Design Space Evaluation of Grid Processor Architectures. In *International Symposium on Microarchitecture (MICRO)*, December 2001.

[6] STANDARD PERFORMANCE EVALUATION CORPORATION. http://www.spec.org.

[7] M. Taylor, W. Lee, S. Amarasinghe, and A. Agarwal. Scalar Operand Networks: On-chip Interconnect for ILP in Partitioned Architectures. In *International Symposium on High Performance Computer Architecture (HPCA)*, Feb 2003.

[8] E. Waingold, M. Taylor, D. Srikrishna, V. Sarkar, W. Lee, V. Lee, J. Kim, M. Frank, P. Finch, R. Barua, J. Babb, S. Amarasinghe, and A. Agarwal. Baring It All to Software: Raw Machines. *IEEE Computer*, 30(9):86–93, Sept. 1997. Also available as MIT-LCS-TR-709.