Versatile Tiled-Processor Architectures: The Raw Approach

Rodric M. Rabbah, Ian Bratt, and Anant Agarwal
Computer Science and Artificial Intelligence Laboratory
Massachusetts Institute of Technology
Cambridge, MA 02139

Recent advances in VLSI technology have created an increasing interest within the computer architecture community to build a new kind of "general 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 compounded by a growing wire delay concern which limits the distance that information can travel in a single clock cycle. The realities of interconnect delay—and power consumption—seriously challenge the ability of microprocessor designers to fulfill the promise of Moore's Law. As a result, new architecture designs are largely centered around scalable and distributed alternatives to current centralized microprocessor designs.

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 architectures that organize silicon resources more effectively and as *tiled-processors* that are easily scalable. 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 IV) 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. One such versatile, tiled-processor architecture (TPA), 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 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 TPAs and architectural versatility necessitates *new benchmark suites and metrics* to accurately reflect the goals of the architecture community. Toward that end, we propose both a new benchmark suite—*VersaBench*—and a new metric called *Versatility*. VersaBench is a collection of applications from three central tiers—desktops, servers, and embedded systems—encompassing traditional integer workloads, floating-point and scientific applications, server computing, stream processing, and bit-level computation. VersaBench thereby attempts to better characterize the broad set of workloads that the new tiled-processor architectures are required to run.

The Versatility 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 (specifically, the VAX 11/780)[1] for each of the applications in the SPEC CINT89 suite. Computing the Versatility of an architecture is purposefully designed to mirror that of SPEC rates. Accordingly, like SPEC, Versatility takes the geometric mean of the speedups of an architecture for each of the applications in the VersaBench suite. Unlike SPEC rates however, the speedup of each application is not computed relative to a single reference machine, but rather relative to the architecture which provides the *best* performance for that application (in the 2004 time frame from known results at the time of this writing).

[1]The reference machines 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.

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 Float	float	medium	medium	medium	medium
Server	integer/float	high	medium to high	medium to high	medium to low
Embedded Stream	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 a diverse set of applications (ranging from desktop to embedded workloads) and multiple forms of parallelism (including instruction-level-parallelism (ILP) for desktop applications, and stream parallelism for embedded computing) as represented by the VersaBench suite. 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. 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 2x-3x better than Raw. Another area of research focuses on desktop integer applications: Raw's performance is 2x lower than a Pentium III for applications with low degrees of ILP.

VersaBench

The VersaBench suite consists of fifteen benchmarks that are grouped into five categories that span desktop, server, and embedded application workloads. The VersaBench applications are themselves drawn from various suites, and were selected because of the salient behavior and the properties they exhibit along various dimensions. For example, it is widely accepted that desktop applications have complex control structures, whereas streaming applications consist of relatively small computational kernels with simple control mechanisms. In all, we consider five property-dimensions when characterizing a benchmark; they are

— *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*

— *data spatial locality*

Intuitively, we believe the properties of each of the five benchmark-categories are as shown in Table 1. Accordingly, the VersaBench suite was created systematically by measuring the properties of numerous applications and selecting those that matched intuition. The presentation will include detailed results that map applications into the five-dimensional space.

Versatility Metric

We define the Versatility of an architecture as the geometric mean of the speedup of every application in the VersaBench suite relative to the architecture which provides the best performance for that application. Thus architectural versatility becomes quantitative, with ASICs (application specific integrated circuits) occupying the lowest end of the spectrum (i.e., Versatility(ASIC) = 0); as future process-technologies deliver higher clock frequencies, architectural versatility will increase beyond unity. Versatility measure servers to identify the areas where opportunities for architectural improvements are greatest, and those where further efforts will lead to marginal returns. For example, a new architecture that performs as well as a Pentium IV for desktop workloads has little to gain from further improvements targeted toward that application domain. In contrast, an architecture that fares poorly compared to the best streaming processor warrants attention that is focused on improving the performance of that architecture within the streaming context.

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.