Versatility and VersaBench: A New Metric and a Benchmark Suite for Flexible Architectures

Rodric M. Rabbah, Ian Bratt, Krste Asanovic, and Anant Agarwal
Computer Science and Artificial Intelligence Laboratory, Massachusetts Institute of Technology

With the increasing miniaturization of transistors, wire delay and power consumption are emerging as the most formidable barriers to the scalability of microprocessors. Overcoming these barriers requires a fundamental rethinking of both microprocessor design and the programming models they support. Toward the former, new architecture designs are focusing on scalable and distributed alternatives to current centralized and monolithic designs. The new architectures are expected to compete with extant microprocessors not in the raw performance they can deliver in specific classes of applications, but rather, in their *versatility*, or the ability to deliver consistently high performance in a very broad set of application domains: server workloads, desktop computing, and embedded systems. The focus on new kinds of "all-purpose" architectures necessitates *new benchmark suites and metrics* to accurately reflect the goals of the architecture community. Hence, we propose both a new benchmark suite–**VersaBench**–and a new metric called **Versatility**.

1) **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 benchmark-selection process should begin with a pool of candidates that exceeded the target number of benchmarks in the suite. In our opinion, the suite should consist of fifteen benchmarks–three benchmarks in each of the five categories, resulting in a manageable suite that will encourage researchers to evaluate the entire suite and not "cherry-pick". For each candidate application, the selection process should not focus on derived measures such as branch prediction accuracy, and data or instruction cache hit/miss rates, but rather on the following fundamental 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 properties of the five benchmark-categories are as shown in the following Table.

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-Level	Bit	Very High	Very Low	Very Low	Very High

Accordingly, the VersaBench suite may be created systematically–by measuring the properties of numerous applications and selecting those that match intuition. Note that while we can generate a similar table using raw data, we believe there are two modes of research that apply here. In one mode, there is significant number crunching that is simply not plausible without substantial time investments, and in the other mode, intuition guides methodology and approach. We subscribe to the second mode of research, and hope that we can motivate companies (e.g., SPEC) to justify the approach or to refute it completely.

2) **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.

The Versatility metric is inspired by SPEC rates. 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) 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 for two reasons. *First*, we believe the geometric mean (GM) 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 GM will increase proportional to the N^{th} root of the speedup. Hence, one application cannot skew the results significantly. *Second*, the SPEC measure is wildly popular and easy to understand, and we do not want to be gratuitously different. Furthermore, it is important to 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 instruction-cache-size.