Summary of Current and Past Research
Parthasarathy Ranganathan

My broad research interests are in the areas of computer architecture, parallel computing, and performance evaluation. My main contributions have been in the areas of (i) architectures for emerging applications (databases, media processing, and communication) and (ii) performance, programmability, and simulation of systems built from processors that exploit instruction-level parallelism (ILP). Below I describe my contributions in greater detail.

Architectures for Emerging Applications (Ph.D. Dissertation)

My dissertation work is motivated by recent trends which indicate that workloads on general-purpose systems have changed dramatically over the past few years, with a greater emphasis on applications such as databases, media processing, and communications. However, in spite of the growing awareness of the importance (and different characteristics) of these workloads, there exists very little quantitative understanding of their performance on general-purpose systems; most architecture studies primarily focus on scientific and engineering workloads. The contributions of my work are:

(i) the first detailed quantitative simulation-based studies of the performance of database and media processing workloads on systems using state-of-the-art processors.

(ii) cost-effective architectural solutions towards achieving the higher performance requirements of future systems running these workloads.

Architectures for Media Processing Applications.

My first study in this area quantified the effectiveness of various features of general-purpose processors for media processing – including aggressive ILP features, media instruction set extensions, software prefetching, and large caches [ISCA99]. My results also showed that (i) media workloads were primarily compute-bound, and (ii) current trends towards devoting large fractions of on-chip transistors for caches (up to 80%) were ineffective for media workloads because of their streaming data accesses and large working sets.

I have proposed and evaluated a new cache organization, called reconfigurable caches, that allows the large on-chip transistor area devoted to caches to be dynamically reused for other processor activities [submitted for publication]. For example, reconfigurable caches could be used to implement (i) look-up tables to reuse memory for computation (e.g., instruction value reuse, value prediction), or (ii) alternative buffer storage for hardware and software optimizations (e.g., prefetch buffers, application-controlled memory). My design requires very few modifications to existing caches and has small impact on cache access times. I believe that this paradigm of reusing on-chip storage for multiple processor activities can facilitate a number of other new architectural optimizations that I plan to pursue further.

Architectures for Database Applications.

In collaboration with Luiz Barroso and Kourosh Gharachorloo at the Compaq Western Research Laboratory, I performed the first comprehensive evaluation of the performance of online-transaction processing (OLTP) and decision support system (DSS) applications for shared-memory servers using state-of-the-art processors [ASP-LOS98]. My main results showed that the key performance-limiting characteristics of OLTP workloads were (i) large instruction footprints (leading to instruction cache misses) and (ii) frequent data communication (leading to cache-to-cache misses in the shared-memory system). Both these inefficiencies could be addressed with simple cost-effective optimizations – stream buffers helped alleviate instruction misses; a combination of prefetching and producer-initiated communication directives helped alleviate the communication misses.

Performance, Programmability, and Simulation of ILP-based systems

My other research work has been motivated by recent processor designs that use techniques such as multiple issue, out-of-order execution, non-blocking reads, and speculative execution to improve performance by exploiting ILP in
the sequential program stream. However, most architectural studies of shared-memory systems assume a more simplistic model of the processor. Together with Vijay Pai, Hazim Abdel-Shafi, and Tracy Harton, I have been the first to explore how ILP features affect the performance, programmability, and simulation of shared-memory servers. Specific results of this work are summarized below.

Performance of ILP-based systems.

ILP features are effective in addressing the CPU component of execution time for our applications, but they are less successful in addressing the memory stall time. Insufficient opportunities to overlap multiple read misses and increased contention for system resources are the key limitations for these systems [HPCA97, IEEEToC99]. Software prefetching is not effective in changing this memory bound nature of most of our applications, mainly due to the large number of late prefetches and increased resource contention with ILP-based systems [ISCA97, IEEEToC99].

Programmability of ILP-based systems.

The memory consistency model of a shared-memory system defines the order in which memory operations appear to execute to the programmer. ILP features can be used to significantly narrow the performance gap between the intuitive sequential consistency model and more relaxed memory consistency models; however, for some applications, a significant gap still remains [ASPLOS96, ASPLOS98, IEEEProc99]. We proposed a new hardware optimization, speculative retirement, to further reduce the performance gap between memory consistency models [SPAA97]. We also proposed and evaluated two new software-controlled techniques, fuzzy and selective acquires, that improve performance by exploiting the additional latency-hiding opportunities provided by non-blocking loads in ILP-based systems [ASPLOS96].

Simulation of ILP-based systems.

At the time we initiated our research, the architecture community did not have access to a simulator that modeled an ILP-based shared-memory system. To address this void, we developed an execution-driven simulator, RSIM, that models the processor pipelines, memory system, and network subsystem of ILP-based shared-memory systems in great detail [TCCA97]. We have made this simulator available in the public domain (with support to the users) and have over 475 world-wide licenses. We have used RSIM to show that commonly used architecture simulators based on simple approximations to ILP processors can be highly inaccurate (with over 150% errors in some cases) [HPCA97].

Other Work

Architectures for Next-Generation Wireless Applications.

I am currently collaborating with researchers from Rice University and the University of Illinois in developing new architectures for next-generation wireless standards. Our initial work has focused on comparing DSP and general-purpose implementations of wireless algorithms and understanding key performance bottlenecks. Our results indicate a need for orders-of-magnitude higher computing performance than those provided by existing architectures [work in progress].

Performance Limits to Future Processors.

I have also analyzed the relative importance of memory latency, bandwidth, and branch predictability on the performance limits of future processors. Using a sampling-based simulation methodology to study a large design space, my results showed that while memory-induced bottlenecks can potentially be hidden in systems with orders-of-magnitude higher capacities to exploit ILP, branch predictability poses a more fundamental limit for both integer and floating-point applications [IRAM97].