

Preface

This issue of the *IBM Journal of Research and Development* contains papers selected from those presented at the 1996 Workshop on Performance Analysis and Its Impact on Design (PAID-96). This workshop was held March 27–29, 1996, at the IBM Austin Research Laboratory, and was sponsored by the IBM Research Division as one of the technical events organized to mark the 50th anniversary of IBM Research. The interest and participation it attracted has led the organizers to repeat it as an annual event.

The PAID workshop series is intended to be a forum for bringing together design and performance architects from industry and academia. The goal is for this workshop to alternate between an academic conference site and an industrial site.

Performance analysis has become one of the critical means of designing well-balanced, efficient processor and system organizations. Such designs must ideally meet or exceed performance targets dictated by the needs of the users for whom they are intended. The “balance” and “efficiency” issues can be related to the overall cost incurred in achieving the performance targets. Hence, the accuracy of pre-hardware projections and design trade-off decisions can be a strong determinant of the commercial success of the actual product. However, since “time-to-market” is an important factor in the success equation, the speed at which the pre-hardware analysis and decisions take place is also quite important. This results in a fundamental speed–accuracy trade-off between the level of detail incorporated in the processor or system model and the accuracy of the attendant analysis.

First-generation computers were organizationally simple, with strictly sequential processing of instructions and little or no overlap of the individual steps. For this reason, architectural performance analysis of such early systems was a straightforward accounting of the number and type of instructions executed and the execution time (or cost) associated with each instruction type. With the rapid advances of the underlying technology, however, computer architects have incorporated higher and higher degrees of overlap and concurrency into their designs. While innovations in architecture and implementation have resulted largely from intuitive insight, it has become increasingly difficult to quantify the benefit of added features. In some product families, this may have resulted in architectural extensions and organizational features of dubious benefit to overall performance. This is so despite early knowledge about how the total system performance is affected by adding a feature which enhances only a given fraction of the whole. (The equation guiding this behavior is popularly known as Amdahl’s Law). As system complexity increases, and the application workloads increase in size, it becomes harder to estimate the base performance, the fraction enhanced, or the speed-up

factor for the feature under consideration. Thus, in conjunction with increases in the complexity of the designs themselves, developers have typically been forced to increase their investment in performance modeling and analysis. This was true in the large systems which evolved in the mainframe segment of the industry, and the phenomenon has repeated itself in the world of microprocessor-based systems.

The difficulty associated with accurate pre-hardware assessments of modern designs is broadly twofold: (a) accurate and timely modeling of the processor or system organization and (b) consideration of workloads and application mixes that accurately reflect the usage of a machine which is to be marketed some time in the future. (Note that the second problem was a legitimate issue even during the earlier generations of computer design.) Even if the absolute projections of performance are not fully accurate, it is important to be correct in making relative trade-offs during the design phase. In any case, having a validated methodology with an “acceptable” level of accuracy allows decisions to be made without relying on designer intuition alone. Moreover, if the model is adequately parameterized, it can be used by experienced architects to conduct experiments which actually foster new ideas for exploration.

Modern machines, which are programmed in high-level languages and incorporate features such as multiprogramming, are not hardware engines alone; rather, they function as a complex interplay among hardware, software, and possibly firmware. In particular, the support provided by the operating system and language compilers often brings up issues involving the effectiveness of functions implemented in hardware or software. For example, even if a compiler is designed or tuned *after* the hardware is implemented, the efficiency of the compiler can have a significant effect on performance. Similarly, effective job-scheduling and resource-management mechanisms are key to reducing system response time. Such mechanisms are usually part of general operating system (OS) functions; however, they can also be incorporated in user applications (e.g., in parallel systems, supporting threaded applications) via explicit directives. Consequently, performance evaluation methods directed at these higher-level interfaces (application–OS, compiler–hardware, etc.) are also important in improving overall system performance.

Another important aspect of developing a robust performance methodology is investing in post-hardware performance measurements. Modern processors and systems often have built-in performance-monitoring hardware to facilitate such measurements. Calibrating the pre-hardware models against actual delivered performance enables architects to refine their existing methodology and to be better equipped for future designs. Also, hardware

instrumentation enables the design team to gather valuable execution traces, which may be used to drive newer simulation models.

This issue of the *IBM Journal of Research and Development* contains a variety of papers which address all of the above aspects of performance analysis. The paper by Kaeli et al. addresses the evaluation methodology for a promising system paradigm which is gaining acceptance in the server community—the CC-NUMA (cache-coherent nonuniform memory access) multiprocessor machine paradigm. The authors describe an AS/400-based prototyping study, conducted with the goal of improving multiprocessor scalability for commercial workloads.

The paper by Emma is a discourse on the essentials of performance analysis as they relate to making the fundamental choices and trade-offs in processor design. The author's development hinges on the key thesis of separable components of the performance equation. He elucidates why cycles per instruction (CPI) is the preferred metric in quantifying architectural performance, as opposed to its inverse (IPC), which has been gaining prevalence. The paper explains a number of what the author believes to be popular misconceptions in proposing newer machine paradigms and their associated analyses.

Aspects of modern-day high-performance compiler optimizations and their impact on overall performance are covered in the paper by Sarkar on high-order transformations. He describes high-level optimizations implemented in the IBM ASTI optimizer, which has been transferred to IBM Toronto for possible incorporation in future high-end machine compilers.

The paper by Charney and Puzak presents a wide range of performance statistics quantifying the behavior of the SPEC95 benchmark suite. The paper is focused on memory system behavior, including the effect of hardware prefetch mechanisms. SPEC95 is the latest benchmark suite established by the Standard Performance Evaluation Corporation (SPEC), a consortium of several hardware manufacturers; it is widely used in analyzing and comparing competing microprocessor products. This is the first in-depth exposition of the characteristics of these benchmarks from the perspective of generic memory hierarchies, with and without prefetch modes, and should serve as a useful reference for practicing architects in research and development.

Moreno et al. present a simulation methodology (with projected results) to model a proposed very large instruction word (VLIW) machine: a modern execution paradigm which has yet to find commercial success in the general-purpose processor marketplace. Such machines rely heavily on sophisticated compiler technology to statically schedule large groups of independent operations to execute in a single processor cycle. Thus, much of the complexity of wide-issue, dynamic superscalar machines is

moved into software. Accordingly, the simulation environment requires the VLIW packets emitted by the compiler to be reinterpreted as native scalar instructions, which are executed and timed (in VLIW time steps) on an existing superscalar machine. This description brings out some of the challenges of accurately assessing the performance potential of a radically new approach by using an existing platform as the simulation host.

The paper by Moreira and Naik deals with evaluation and optimization at the application interface. They enunciate the concept of dynamically reconfigurable applications in the context of an IBM SP2 parallel hardware platform. Their evaluations point to novel, user-directed mechanisms to increase overall system utilization and to reduce job response time.

In their paper, Sandon et al. address an important aspect of post-hardware measurement: the ability to capture representative instruction traces from an existing system, running real workloads. They describe NStrace, a bus-driven tool for deriving such workload execution traces.

Another treatment of post-hardware evaluations is given by Levine and Roth. They describe the performance-monitoring facility provided in the PowerPC 604 microprocessor. Such monitoring facilities (if available) have tended to be largely proprietary in prior mainframe and high-end microprocessor-based systems. Consequently, publications on the programming interfaces, hardware mechanisms, and actual measurements related to such facilities in real systems are not easy to find. This paper bridges part of that knowledge gap, at least for a specific product family.

It is not possible to capture in fewer than ten papers the complete range of topics addressed at the PAID-96 workshop. I believe that some of the most important ones are here, however, and I appreciate very much the cooperation of the authors, who spent much time in preparing final versions for publication. The editors would also like to thank the many reviewers who took the time to prepare the detailed reviews which enabled the authors to bring out fully the high quality of the papers in this issue.

Pradip Bose
(PAID-96 Workshop Chair)

Guest Editor