030.matrix300 Performance Considerations

By Bill Keatts (Control Data), Subra Balan (IBM Corporation),
Bodo Parady (Intergraph Corporation)

Introduction

030.matrix300 is one of the six FORTRAN programs in the SPEC Release 1 benchmark suite used to calculate SPECfp and SPECmark measures. Recently, several vendors have reported significantly higher SPECfp and SPECmark results. The before and after results and the factor of improvement for some of the systems are listed in Table 1 (below).

This situation has generated significant concern in the industry and has led to many questions and comments from the press and analysts about whether the benchmark is broken, whether these vendors have specifically optimized their systems to run this benchmark well and what this means to the customer. This article addresses these concerns.

Reasons for the Dramatic Performance Improvement

030.matrix300 was accepted into the initial SPEC Release 1 suite to represent matrix operations frequently seen in FORTRAN applications. Since it was difficult to obtain real application codes due to their proprietary nature, the frequently-used subroutine, "saxpy," was selected to represent 030.matrix300 operations.

The dramatic performance improvements in 030.matrix300 are a result of incorporating either in a preprocessor or integral to the compiler, the optimization algorithms developed by Kuck & Associates. The KAP preprocessor from Kuck & Associates incorporates these algorithms. These techniques have been developed through research on real application codes over the last 20 years. Some of these optimization techniques are :

Sub-program inlining - process of replacing a subroutine CALL with the actual text of that subroutine,
Memory management - optimization of code for efficient use of memory and cache,
Loop reordering - reorder nested loops to make most efficient use of cache,
Dead code elimination - variables are analyzed and when not used again, they are considered dead code and eliminated.

The performance improvement varies depending on the architecture and floating point performance of the system.

So, Is It a Trick?

The answer to whether the optimizations have been made specifically for this benchmark is "NO," for the following reasons :

- Techniques to improve algorithms represented by "saxpy" had been developed well before SPEC benchmarks were available and were therefore not targeted at just improving SPEC results. Just recently, these techniques have been incorporated into vendor software to enhance performance.

- The compute intensive performance of a system is dependent on the hardware and the compiler technology. While

Table 1: 030.matrix300 performance

System	030.matrix300 Performance (SPECratio)		Improvement Factor
	Without KAP Preprocessor	With KAP Preprocessor	
CDC 4680	50.8	163.9	3.2
HP 9000 730	36.1	415.1	11.5
IBM RS/6000 550	78.3	729.8	9.3
Sun SPARCstation 2	28.1	82.6	2.9

Note: Please refer to the Benchmark Terminology section on page 22 for definitions of SPECfp, SPECmark and SPECratio.

quantum leaps in performance due to hardware enhancements are accepted, similar gains due to advancements in compiler technology are misinterpreted as "defeating the benchmark."

- The improvements in 030.matrix300 performance are not bad or tricks of benchmarking, as has been proposed in some articles. The KAP preprocessor significantly reduces memory traffic by reducing cache misses. Each cache miss avoided saves between 10 and 20 instruction cycles depending on the architecture. These optimization techniques from Kuck & Associates are certainly aimed at improvements in real application performance and accomplish this goal. From the standpoint of generally improved performance for customers, more and more techniques such as those employed by the KAP preprocessor are needed and will continue to be developed. To the extent that benchmarks represent real production application usage, improvement in the benchmarks will foretell improvements in customer applications as well. In other words, these improvements will allow customers to obtain additional benefit from hardware products they purchase.

What About the Future?

In the SPEC CPU Benchmark suite planned for release early in 1992, the 030. matrix300 program will be removed in favor of other programs which more closely represent matrix applications in use within the industry. The reasons for this follow.

- "saxpy" was an important kernel on early single CPU vector computers that had no cache but it is no longer fundamental to computer system performance. This has been recognized by others including Jack Dongarra, the author of the Linpack benchmark and the BLAS level 1 routines, of which "saxpy" is an important component. Due to the development of far more aggressive architectures, including parallel and parallel vector computers, the fundamental unit of floating point computation has moved from BLAS level 1 (vector-vector operations) routines, to BLAS level 2 (matrix-vector) routines, and finally to BLAS level 3 routines (matrix-matrix) routines. This is reflected by Jack Dongarra in his latest benchmark, which attempts to measure the ability of a computer to multiply a large 1000x1000 matrix for double precision floating point arithmetic.

- In the case of 030.matrix300, an extremely aggressive compiler could recognize that there was no input data and thereby construct the answer through constant propagation at compile time. All benchmarks in the new floating point suite (with the exception of tomcatv, which is not susceptible to constant propagation) will eliminate this by using input data from a file rather than using compile time constants.

In order to set the record straight on the 030.matrix300 situation and to point out a number of issues with regard to performance benchmarking, the following points need to be made:

- Software optimization is necessary, and the trend which has existed since compilers were first developed, of optimizing for improved performance, must continue. Typically, hardware architectures between the time of introduction and phase-out may show at least a two times performance improvement simply as a result of improved software. Achieving higher system performance by software optimizations on the same hardware is necessary to approach maximum theoretical performance for a given architecture.

For example, the performance improvements realized by hardware enhancements and by software optimization are contrasted below:

Table 2: 030.matrix300 performance

System	Clock (MHz)	030.matrix300 SPECratio
Sun SPARCserver 490	33	24.6
Sun SPARCstation 2 (without KAP preprocessor)	40	28.1
Sun SPARCstation 2 (with KAP preprocessor)	40	82.6

Using Table 2 (above), the 030.matrix300 performance improvement due to hardware enhancement (from 33 MHz to 40 MHz SPARC) is a factor of 1.14 (28.1 divided by 24.6). And, the 030.matrix300 performance improvement due to software optimization (KAP preprocessor on the SPARCstation 2) is a factor of 2.9 (82.6 divided by 28.1).

Similar computations may be made for other vendor systems. Please be aware that software optimizations may not always provide such dramatic performance improvements. However, we have shown that performance is more than just hardware. It is based on the hardware architecture and speeds, operating system and compiler algorithms, disk and terminal I/O capability and application design and coding. And, software optimization enhancements may provide higher performance in the technical computing environment rather than the commercial due to the fact that commercial environments require use of more of the system's performance components than just the hardware (CPU) and compilers.

Benchmarks, and particularly benchmark standards, by their very nature will be the target of architectural and software optimization advances. Thus, a significant weight of responsibility lies on the shoulders of those involved in benchmark standards development. This responsibility is felt strongly within SPEC and is viewed as an opportunity to influence architectures and optimizations toward overall improved performance on real customer workloads.

Conclusions

Several lessons can be learned from the 030.matrix300 experience.

- As technologies change, the manner in which users code programs will also change. At times, the new technologies negate the usefulness of benchmarks. This happened to Whetstone and Dhrystone. In the case of Whetstone, spurious dummy subroutine calls were optimized away. And, in Dhrystone, procedure integration techniques nullified the attempt to time subroutine calling, a major portion of the program. Additionally, these programs fit entirely in the cache of today's workstations leading to higher performance than a user's program would realize. Therefore, SPEC and other benchmark development groups must continually review benchmarks to ensure that the type of work done in the benchmark is representative of the industry. In the case of 030.matrix300, SPEC will achieve this by replacing it with other programs.

- The drive for improved performance will be addressed not just by hardware, but by software and application algorithms in order to achieve the performance demands of our customers.

- Vendors and customers should continue to expect improvements in performance measures for any benchmarks that are used for comparison. Standard benchmarks will continue to be targets for improvement under the assumption that such improvements will be represented by improvements in customer production results.

References

SPEC Newsletter , Volume 3, Issue 3, September 1991
SPEC Newsletter, Volume 3, Issue 2, Spring 1991
SPEC Newsletter, Volume 3, Issue 1, Winter 1991
HP Apollo 9000 Series 700 Performance Brief, March 1991

Bill Keatts is director of Performance Analysis at Control Data's Computer Products Group in Minneapolis, Minnesota. Subra Balan works at the IBM Performance Evaluation Center located in Roanoke, Texas. Bodo Parady is a senior systems consultant at Intergraph Corporation's Advanced Processor Division, located in Palo Alto, California.

SPECtacular *(continued from page 1)*

Network Benchmark Suite

SPEC was approached in mid 1991 by the LADDIS group about the possibility of SPEC being the group responsible for introduction of their version of the NHFStone networking benchmark. LADDIS was formed by representatives of Legato, Auspex, Data General, DEC, Interphase, and Sun Microsystems. Since three members of LADDIS (DEC, Data General, and Sun) were also represented in SPEC, there was a natural inclination to seek this alliance. SPEC gladly accepted this challenge, and members of other SPEC firms are now involved in porting this benchmark to their platforms.

Siemens Nixdorf has also offered SPEC a network benchmark that is currently being completed at their headquarters in Germany. This benchmark has the added advantage of being written to the demanding DIN standard.

It is highly likely that SPEC will introduce a network benchmark suite during 1992. Other network companies have been approached, and been asked to bring their in-house network benchmarks and expertise to our growing Performance Cooperative.

1992 Will Be a Great Year

Although SPEC may not be successful in its efforts to introduce all four of the benchmark suites discussed above during 1992, it is certain that the release pace will be far beyond the one benchmark suite per year pace of the past.

SPEC continues to invite all interested parties to join in the effort to develop fair and effective tools to measure all aspects of computer system performance.

Bud Funk is the chairman of the Steering Committee of SPEC and a project engineer with the UNIX Systems Group of Unisys, located in Salt Lake City, Utah.