fission energy from the public-acceptability point of view.

It also follows from the present study that there is no more any justification for pursuing exotic schemes for the neutron transmutation of 'waste actinides' using novel non-fission neutron sources such as spallation targets, superconducting cyclotrons or fusion-reactor blankets, for it is obvious now that all these nuclides are, if at all, better fuels for any type of fission reactor than even plutonium.

The study also has a bearing on the so-called shelf-life of plutonium. There seems to be a widely accepted contention that when 'fissile' ^{241}Pu (half-life of 14.4 years) decays to 'non-fissile' ^{241}Am there is a 'significant loss' of fuel value[12]. In fact this 'belief' has often been used to justify the near-term recycle of the growing stocks of plutonium in LWR while awaiting the commercialization of liquid metal fast breeder reactors (LMFBR), which has been delayed. The systematics of criticality data presented in this paper (see Table 1) clearly indicates that the fissionability properties of ^{241}Am with $K_\infty = 2.519$ and $\sigma_c^b = 45.39$ g cm^{-2} (Table 1 column 9), seem to be better than those of even ^{235}U, whose K_∞ is 2.337 and σ_c^b is 51.53 g cm^{-2} in a hard-spectrum fast reactor.

A study of systematics of the type reported here provides the possibility of identifying those nuclides whose basic nuclear data warrant reexamination. For example, the data points of ^{247}Cm and ^{231}Pa, which fall outside the overall trend, possibly warrant a reappraisal of their input nuclear data. Further, by scrutinizing whether the deviation from the overall trend occurs in the $\bar{\nu}$ plot, the K_∞ plot or the $1/\sigma_c^b$ plot, one can assess which segment of the nuclear data needs refinement.

1. Srinivasan, M., Subba Rao, K. and Dingankar, M. V., Proceedings of ANS/NIST Conference on 50 Years with Nuclear Fission, Gaithersburg, USA, 1989.
2. Chapline, G., *Nucl. Instrum. Methods*, 1988, **A271**, 207.
3. Clayton, E. D., Proceedings of International Seminar on Nuclear Criticality Safety, Tokyo, Japan, 1987.
4. Kelley, R. E. and Clayton, E. D, *Nucl. Sci. Eng.*, 1985, **91**, 481.
5. Westfall, R. M, *Nucl. Sci. Eng.*, 1981, **79**, 237.
6. Srinivasan, M., Subba Rao, K., Garg, S. B. and Acharya, G. V., *Nucl. Sci. Eng.*, 1989, **102**, 295.
7. Garber, D. *et al.*, 'Data Formats and Procedures for the Evaluated Nuclear Data File, ENDF', Brookhaven National Laboratory Report BNL-NCS-50496 ENDF-102, 1975.
8. Howerton, R. J., 'The LLL Evaluated Nuclear Data Library (ENDL)', UCRL-5400, vol. 15, rev. 1, Lawrence Livermore National Laboratory, USA, 1982.
9. Garg, S. B. and Sinha, A., 'BARC-35, A 35-BARC cross-section library with P3-anisotropic, scattering matrices and resonance self-shielding factors, BARC-1222, Bhabha Atomic Research Centre, Bombay, India, 1984
10. Manero, F. and Konshin, V. A., *At. Energy Rev.*, 1972, **10**, 637.
11. Indo-Japan Seminar on Thorium Utilization, Bombay, India, 1990
12. Van Dievoet, J. Proceedings of ENC-3 Conference on New Directions in Nuclear Energy with Emphasis on Fuel Cycles, Brussels, Belgium, 1982.

Received 24 May 1990; revised accepted 3 April 1991

RESEARCH COMMUNICATIONS

The role of compilers in computer-system performance

K. Neelakantan, G. Athithan and P. P. Ghosh

ANURAG (Advanced Numerical Research and Analysis Group), P. O. Kanchanbagh, Hyderabad 500 258, India

The compiler is a system code which translates the user's program into machine-executable code and thereby dictates the performance of the computer system. Here we present results obtained while benchmarking various FORTRAN compilers on Motorola 68020-based machines. Our results dramatically show the effect of compilers on the performance of computing systems.

THE Advanced Numerical Research and Analysis Group (ANURAG) is building a parallel-processing system called PACE[1], primarily for computational-fluid-dynamics (CFD) applications. CFD is computationally intensive and therefore the basic aim of the parallel-processing system is to provide a platform that gives very high computational throughput. This requires careful choice of hardware. The hardware and operating-system environment of PACE have been chosen on the basis of several considerations (reported elsewhere, see ref. 2). For a given system configuration, the performance hinges crucially on the efficiency of the compiler used to generate executable code.

Our purpose in this paper is to report some results and observations made while benchmarking several FORTRAN compilers for use on our parallel processor. The study was motivated by the need to maximize computational throughput on PACE. Generally one associates system speed with hardware performance. However, the systems software also has a significant role in determining system speed. The benchmark results reported in the literature do not usually consider this aspect. Therefore, having chosen the specific system and hardware configuration for PACE, ANURAG decided to explore the possibility of fine-tuning the performance by appropriate choice of compiler.

The microprocessor used in PACE is the Motorola MC680 × 0. The operating system in the host processor is UNIX while the nodes run under the control of a multi-tasking real-time kernel[1]. The nodes do not have any system resources other than memory and communication channels. The number-crunching is done at the nodes. The front-end processor does not usually participate in the computations. Therefore, for any given applications code, the performance of the system is dictated by the choice of the compiler.

The system supports several programming languages but FORTRAN is considered to be the most important of these as most (CFD) users seem to prefer FORTRAN. In order to obtain the highest possible computational throughput, ANURAG wanted to choose an appropriate FORTRAN compiler. It was therefore decided to benchmark several FORTRAN compilers that were commercially available to select one for use on PACE.

Benchmarking systems is full of pitfalls[3] and each benchmark program has its limitations. Since PACE is to be used for number-crunching applications, we decided to choose three programs that best exercised floating-point computations and are therefore representative of scientific computation.

The first program is the familiar Whetstone[3] benchmark. This is a general-purpose benchmark that exercises the basic floating-point functions such as add, subtract, multiply, divide, as well as trigonometric functions. The results are reported as Whetstone KIPS (kilo-instructions per second), and are a fair measure of the overall- performance of the system in a general scientific-computing environment.

The second program is the Linpack benchmark[1,3,4], which solves a system of linear equations. This is representative of the programs used to numerically solve partial-differential equations. Since PACE is to be used in CFD applications, this is an appropriate benchmark.

The third program is the computation of fast Fourier transforms (FFTs). FFT is frequently used in scientific computations and requires floating-point computations on complex numbers. We restricted our measurements to 4096-point FFTs using the FORTRAN subroutine given by Cooley, Lewis and Welch[5]. The FFT benchmark essentially excercises the capability of the system to handle computations on complex numbers.

All the benchmark programs were run under a UNIX environment (or a UNIX-like environment). The times were obtained using a standard UNIX call 'clock'. The benchmarking was carried out on several compilers. These are briefly described below.

(i) The UNIX FORTRAN compiler: this is not an efficient compiler as it generates an intermediate C code, which is later compiled by the C compiler under UNIX.

(ii) The ECIL FORTRAN compiler: this compiler is a reasonably efficient compiler developed indigenously by the Electronics Corporation of India (ECIL).

(iii) The Softek FORTRAN compiler is an indigenously developed optimizing compiler.

(iv) The Greenhills FORTRAN is a popular optimizing compiler.

(v) The Silicon Valley Software (SVS) FORTRAN: this is again a very popular compiler with optimizing options.

(vi) The SUN FORTRAN is used on the SUN workstation and has optimizing options.

The results of the performance measurements are given in Tables 1 and 2. In Table 1, we present the performance of the benchmarks run on ECIL's UNIPOWER-20 machine. This machine is based on the Motorola MC68020 with the MC68881 coprocessor running at a clock speed of 16.67 MHz. The operating

Table 1. Benchmark results for various compilers on the UNIPOWER-20.

Salient system specifications:
Motorola MC68030 CPU with MC68881 coprocessor
Clock speed 16.67 MHz
1 MB RAM

Compiler	Whetstone KIPS	Linpack MFLOPS	FFT (4096-point) complex KFLOPS
UNIX FORTRAN* without 68881	32.3	0.00687	2.18
Softek FORTRAN*	1145.1	0.077	32.8
ECIL FORTRAN†	362.3	0.033	2.93
Greenhills FORTRAN†	890 5	0.095	25.21

*UNIX V.2 operating system
†UNIX V.3 operating system

Table 2. Comparison of various compilers on MC68020-based machines.

Machine	Compiler/OS	Whetstone	Linpack	FFT
OMEGA IRIS 68020/Weitek at 16.67 MHz	SVS FORTRAN UNIX V.3	2069.0	0.315	25.2
OMEGA IRIS 68020 without coprocessor at 16.67 MHz	SVS FORTRAN UNIX V.3	234	0.033	8 13
Apollo DN3000 68020 + 68881 at 12 MHz	UNIX FORTRAN DOMAIN IX	409 6 (569 0)	0 031 (0 043)	7.56 (10 50)
SUN 3,60 68020 + 68881 at 16 67 MHz	SUN FORTRAN UNIX V.3	774	0 0857	14 9
DCM DUAL II 68020 + 68881 at 20 MHz	SVS FORTRAN	980 (816.8)	0.100 (0 083)	–

The figures in parenthesis give the performance normalized to a 16 67-MHz frequency of operation.

system was UNIX V.2 (UNIX V.3 was also available), and the system has an on-board memory of 1 megabyte (MB). Our emphasis on this machine is due to the fact that the processor boards used in UNIPOWER-20 are identical to the ones used in the initial versions of PACE.

Initially, the studies were conducted on the UNIPOWER-20 only. However, the results of these benchmarks were so interesting that we extended the study to other compilers and systems as well. In Table 2 we present results obtained on some other MC68020-based machines. These systems had slightly different hardware configurations and clock speeds but are all based on the MC68020 processor. Sufficient memory was available to ensure that the programs could be run without recourse to virtual storage. The point is that, though the details of the hardware configuration are different, the performance is dictated mainly by the efficiency of the compiler and of course the clock speed. It would have been ideal if all the compilers could have been ported to a single machine and the benchmarking done on the same hardware and operating system. However, this was not possible owing to certain practical difficulties.

The results in Table 1 clearly show the dominant role the compiler plays in system performance. The Linpack rating on the UNIPOWER-20 with the ECIL FORTRAN compiler is 0.033 MFLOPS (million floating-point operations per second) while with the Greenhills FORTRAN compiler it is 0.095 MFLOPS. The ratio of approximately 3 is significant in the sense that it can easily swamp out the advantages of a superior hardware configuration.

A careful look at Tables 1 and 2 would show that the ratios of performances of one machine over another are *not* invariant across the different benchmarking programs. In Table 1, for instance, between Softek FORTRAN and ECIL FORTRAN, the Whetstone-rating ratio is about 3.5, whereas the Linpack-rating ratio is about 2.3. The Softek compiler performs better than the ECIL compiler because the former is a highly optimizing compiler. As the scope of optimization is rather more in non-floating-point operations such as conditional branching, accessing of arrays, etc., the Whetstone ratings, which evaluate the overall performance of a system, including floating-point performance, come out better than Linpack ratings in the case of a highly optimizing compiler. Complex floating-point operations are in a different class and special care is required in carrying them out. One can see that the Softek FORTRAN performs better than the Greenhills FORTRAN in the FFT benchmark even though the Greenhills compiler gives a better Linpack rating. In fact the Softek compiler running on a UNIPOWER-20 which uses a MC68881 coprocessor (Table 1) outperforms the IRIS workstation (with an SVS compiler) on the FFT benchmark (Table 2). This is interesting because the IRIS workstation has a special floating-point accelerator (Weitek 1164/1165), and a Linpack rating of 0.315 MFLOPS. The hardware advantage is clearly being swamped out by compiler performance.

In Table 2, one may note that both SUN FORTRAN and SVS FORTRAN have comparable and high ratings, while UNIX FORTRAN in Apollo DN3000 comes out second best. In all three cases, the ratios of the ratings for different compilers, viewed across the different benchmarking programs, are within reasonable limits.

These results emphasize the importance of the compiler in determining system performance. A poor compiler can more than swamp out the advantages of a superior hardware configuration and a faster processor. In fact we attribute the excellent results[1] we obtained with PACE-8 to the fact that we have used an optimizing compiler that extracts the maximum performance out of the hardware. This partially explains the reason why PACE-8 is able to deliver performance comparable to that of machines with better stated performances. For example, Transputer is supposed to be a processor capable of delivering over 2 MFLOPS of computational speeds as measured by its hardware performance. However, its Linpack rating is merely 0.42 MFLOPS[6]. Part of the reason for this drop in performance could be the absence of a good optimizing FORTRAN compiler for Transputer-based systems.

In this paper we have used several names that are trade marks and vendor names. This is inevitable while describing compilers and systems. However, we do not mean to imply that any particular commercial product is superior to others. The results of the benchmarks must be taken in the spirit of the paper, which is to emphasize that the compiler is important in determining floating-point performance. Obviously, many other parameters must be taken into account while comparing commercial products.

1. Neelakantan, K., Ghosh, P. P., Ganagi, M. S., Athithan, G., Atre, M. V. and Venkataraman, G., *Curr. Sci.*, 1990, **59**, 982.
2. Neelakantan, K. and Athithan, G., 'PACE user information', ANURAG Report No. ANU/PACE/89/01, Hyderabad, India, 1989.
3. Wilson, P., *Byte*, March 1988, 217.
4. Dongarra, J. J., 'Performance of various computers using standard linear equations software in a FORTRAN environment', Argonne National Laboratory Technical Memorandum No. 23, Argonne, USA, 1989
5. Cooley, J. W, Lewis P. and Welch, P. D., *IEEE Trans. Audio Electroacoust.*, 1967, **15**, 76.
6. Dongarra, J. J., 'Performance of various computers using standard linear equations software', University of Tennessee Report No. CS-89-85, Knoxville, USA, 1990.

Received 4 April 1991; accepted 1 May 1991