A Study of Workstation Computational Performance for Real-Time Flight Simulation

Jeffrey M. Maddalon
Jeff I. Cleveland II

Summary

This paper presents the results of a computational benchmark, based on actual real-time flight simulation code of an X-29 aircraft used at Langley Research Center. This benchmark was run on workstations from Digital Equipment Corporation, Hewlett-Packard, International Business Machines, Silicon Graphics, and Sun Microsystems. The intent of this study is to measure the computational suitability of workstations to operate a simulation model of an X-29 aircraft. Before computational performance can be considered relevant, computational accuracy and software porting costs must be found to be acceptable. This study indicates that, in general, workstation vendors have no problem meeting computational accuracy requirements for the X-29 aircraft simulation model. Porting this simulation model to different computational platforms shows that there is little middle ground, porting will be either easy or difficult. The computational performance results show that workstations from several vendors can provide the necessary computational power (along with sufficient computational accuracy and moderate software porting costs) to properly operate a real-time flight simulation model of an X-29 aircraft.

Introduction

In the past, mathematical computations performed during real-time flight simulation at Langley Research Center have required the high-performance floating point processing of supercomputers. With recent advances in microprocessor technology, some have suggested that modern workstations provide enough computational power to properly operate a real-time simulation. This paper presents the results of a computational benchmark for real-time flight simulation which was executed on various workstation-class machines. The results presented in this paper are from a single program (the benchmark), other programs may have different performance characteristics and different conversion difficulties.

The advantage of supercomputers instead of workstations is performance. The benefits of using workstations instead of supercomputers are numerous. The principle benefit is the lower initial cost of workstations as compared to supercomputers. Compared to supercomputers, a large number of workstations are in use; so, errors in software tend to be found sooner than software problems on supercomputers. Because of reduced hardware complexity, workstation hardware tends to be more reliable than supercomputers. With this increased reliability, maintenance costs on workstations are considerably less expensive than on supercomputers. Finally, the low initial cost of workstations allows cost-effective machine redundancy which further increases reliability.

This study is intended to measure computational performance of the machines tested and not the

full range of characteristics needed for proper real-time operation (e.g., input/output performance and interrupt response time). Some machines had features which could increase performance, but the use of such features may interfere with the real-time operation of the machine (e.g., parallel processing); these features were not used. In addition, the performance of a machine is irrelevant if the machine does not produce correct answers or if the cost to move the benchmark to the machine is exorbitant--these factors were also measured.

The benchmark was executed on different machines from several companies including: CONVEX, Cray Research, Digital Equipment Corporation, Hewlett-Packard (HP), Intel, International Business Machines (IBM), Silicon Graphics (SGI), and Sun Microsystems. Table 1 lists all the machines tested, along with relevant information about these machines. This computational study was aimed at determining the performance of workstations; however, as the accompanying table indicates, several supercomputers were examined. Since the current simulation environment at Langley uses two CONVEX supercomputers (C3840 and C3820), these systems were tested to gauge the performance of the current simulation system. The Cray supercomputers were tested because those machines are good representations of the most powerful vector supercomputers currently available.

Description of Benchmark

The benchmark is a mathematical model of two X-29 aircraft developed from a real-time simulation of one of these aircraft. The benchmark computes the equations of motion and the control laws of both X-29's. Since this benchmark is derived from an actual real-time simulation, this program is very representative of the computational complexity of current simulations used in Langley's simulation system. The benchmark spends 90 percent of its execution time guiding the planes through a repeated series of pseudo-piloted maneuvers consisting of pitch, roll, and yaw doublets. During the remaining 10 percent of the time, the aircraft are allowed to fly without pseudo-pilot intervention. In a true real-time environment, the benchmark would run for 1,000 seconds with 80 solutions of the mathematical model each second. The time reported by the benchmark represents the amount of time that must be dedicated to computations. The difference between the total time of 1,000 seconds and the reported time represents the time for communication between the simulation computer and the cockpit and the time for other tasks performed during real-time operation.

The benchmark provides two different execution modes: data mode and timing mode. Data mode produces a file containing approximately 10,000 intermediate values. Along with a master file containing manually-verified results, this mode is used to test the accuracy of a machine's computations. Timing mode produces no such data file; it simply reports the elapsed time which is used to determine the machine's performance. Normally, benchmarks report the amount of central processing unit (CPU) time used as opposed to elapsed time (wall clock time). Elapsed time was reported to account for operating system overhead and other machine delays that normally would not be reflected in the amount of CPU time. Using elapsed time does have a significant drawback: only the benchmark may be running on the system. This is similar to how the machines would operate in a true real-time environment. With the exception of the Cray computers, the difference between the amount of CPU time used and the elapsed time was less than 1 percent. Due to the high workload on the Cray machines, elapsed time was not an accurate measurement of

performance, so CPU time was reported for the Cray machines.

The benchmark is a 24,000 line FORTRAN program which uses approximately 1 megabyte of main memory while in operation. Since the program uses such a small amount of memory, no disk paging occurred. The lack of disk paging was verified on all machines that provided the necessary tools to measure paging. Thus, the benchmark only measures processor and memory performance; it does not measure disk or input/output subsystem performance. The execution of the benchmark was strictly limited to a single processor on all machines. Multiple processors were not used for two reasons. First, the benchmark does not break into parallel modules easily. Second, some types of parallel computations place unpredictable loads on the processor which would violate the determinism requirement of real-time systems. The benchmark uses very few vectors, so the vector registers of the Cray and CONVEX machines provide little additional performance. Because of the stringent time requirements of Langley's real-time simulations, long, complex mathematical procedures are not used in the benchmark. Therefore, this benchmark is particularly suited to machines that execute relatively simple instructions very rapidly.

In terms of computational requirements, the benchmark must use extended precision (greater than 48 bits) for all real numbers. For most computers, this translates into 64-bit floating point representation as opposed to 32-bit floating point values commonly found on workstation-class machines. If the extended real values are not used, the cumulative effect of this loss of precision causes the benchmark to end prematurely with a division by zero error. Most of the real values in the code are implicitly defined or simply declared with the "REAL" statement (without a size designation). On machines which do not automatically generate 64-bit real values, a compiler option is needed to force all real values to the larger precision. To meet the extended precision requirements the default version of various intrinsic functions must operate correctly at the extended precision. Because of the ancestry of the program, the benchmark expects subroutine local variables to remain instantiated throughout the life of the program; this feature is sometimes referred to as static local variables.

Machine Results

Benchmark results are summarized in Table 2. The reported time is the execution time of the benchmark, so a lower number means the benchmark executed faster. All times represent the average of at least 10 runs with the exception of the Cray machines where the benchmark was run only once due to the high-computational load on those machines. On all machines, the benchmark was run with the optimization level that gave the fastest speed on that machine while maintaining computational accuracy; however, no optimizations were performed that require a programmer to modify the code to gain additional performance.

The performance index is the machine's performance relative to the CONVEX Computer Corporation C32xx series of computers and is derived by dividing the C32xx execution time by the time for each system. The CONVEX C32xx supercomputer was a computer previously used for model computations in Langley's real-time system and is considered a machine with the minimum performance necessary for Langley's real-time simulation programs.

Any inaccuracies between the machine's results (generated in data mode) and the manually-

verified results from the master file are summarized in the inaccuracy column. The percent difference between the machines results and the manually-verified results (the master file) is less than or equal to the number in the inaccuracy column. The master file was originally created by a Control Data Corporation CYBER 175; thus, the inaccuracy for the CYBER 175 is 0.0 percent. The benchmark was originally developed on the CYBER 175. Inaccuracies of 0.0001 percent or less are considered fully accurate. As the footnote indicates, the results from the Cray-2, the HP 9000-735, and the HP 9000-720 disagreed with the manually-verified results; however, they agreed with each other. This implies that slight differences in the implementation of certain critical functions may cause the benchmark to be sensitive to small differences in one or more computational results. Furthermore, if the benchmark can be slightly modified to eliminate these sensitivities, the benchmark may produce a file that agrees with the master file; however, the execution of the modified benchmark could have different performance characteristics.

The last column indicates the difficulty in moving the benchmark to different machines. A full explanation of this column is given in the next two sections.

Software Portability Rating System

Software portability indicates how much effort was required to move the benchmark to the specific hardware platforms. This rating system attempts to objectively gauge the difficulty of porting software to the various platforms.

The difficulty of the porting effort was rated in three categories: easy, moderate, and difficult. The rating was determined by the amount and types of problems encountered while moving the benchmark to the target computer. Each problem was put in one of three categories: annoyance, significant, and serious. An easy rating means no serious or significant problems, and less than three annoyance problems. A moderate rating means no serious problems and less than three significant problems, or three or more annoyance problems. A difficult rating means one or more serious problems, or three or more significant problems. Specific problems with their category are given below.

Serious problems interfere with the operation of the computer and need to be addressed by the vendor. Whenever a compiler does not operate in accordance with the American National Standards Institute (ANSI) standard FORTRAN, a serious problem is recorded. The problem of a machine that does not give the correct answers from a function or operator is also rated as serious. Problems with computational accuracy require detailed and time-consuming code traces; thus, these problems are rated serious. One exception to this is if the documentation for the compiler indicates that a specific compiler option may cause computational inaccuracies.

Significant problems are problems that should be addressed by the vendor, but do not seriously interfere with the operation of the system. Run-time errors not related to computational accuracy are rated as significant problems. Incorrect documentation is rated significant. If the compiler does not provide the necessary options (e.g., options for extended precision and static local variables) the rating is significant.

Annoyance problems are the least serious problems. Generally, the system reports the cause and

location of an annoyance problem and a work-around can be found easily. Errors found at compilation-time are rated as annoyance problems. The absence of routines that are not in the ANSI standard but are commonly available (e.g., time and date routines) is rated an annoyance problem. Unclear documentation is rated as an annoyance problem.

Software Portability Results

The experience of moving the benchmark to machines in this study is presented below. The benchmark was originally built for the CDC CYBER machines and was previously converted to the CONVEX machines so no conversion effort was required for either of these machines. Thus, the porting effort for these machines is not indicated in Table 2.

The benchmark results from the Cray Research computers were incorrect. This problem is rated serious. High machine workloads prohibited investigation of the problem. Since the one problem encountered during the porting to the Cray computers was rated serious, the total effort was rated difficult.

Converting the benchmark to the Digital 3000 models 500, 500X, and 800 was simple. The benchmark compiled almost without problems--the compiler indicated one function that would not operate with the extended precision (an annoyance problem). The documentation for the compiler is excellent: the information is both clear and useful. The one annoyance problem means the rating for porting the benchmark to the Digital machines was “easy.”

Many problems were encountered converting to the HP machines. The first problem arose because a compiler option to automatically use 64-bit floating point values does not exist (a significant problem). A hand conversion effort did allow the benchmark to run. Several problems were encountered with the equivalence statements in the benchmark (compile-time, annoyance problem). The most important problem was that one statement gave the wrong answer to an arithmetic operation when a comment was placed in column 73 (a serious problem). In addition to these problems, the answers given by the HP machines did not agree with the true results (another serious problem). The two problems rated as serious made converting the benchmark to the HP machines difficult.

The porting effort to the IBM personal computer clone (using the Intel i486 microprocessor) running a UNIX operating system was difficult. These difficulties were mainly caused by problems with the Edinburgh Portable Compilers FORTRAN 77 compiler. One problem was isolated to a computational inaccuracy with nested statement functions (a serious problem). Other problems include: under certain conditions, statement functions incorrectly return an indefinite value (a serious problem) and the compiler occasionally uses previous versions of certain routines between different compilations of the program (a serious problem). By coding around these problems, the benchmark did return the correct values.

The IBM computers had a few problems during conversion. The compiler option to automatically use 64-bit real values was somewhat cryptic, but once exercised, it worked properly (unclear documentation--an annoyance problem). The IBM machines do not have certain timing subroutines available on other machines (two annoyance problems). Standard UNIX utilities

compensated for this deficiency. With the three annoyance problems, the porting effort to the IBM systems was moderate.

Conversion to the SGI machines running version 4.0.5, 5.0, or 5.1.1.2 of IRIX was difficult. Extended precision incompatibilities in library subroutines caused several problems which could be corrected by changing the subroutine from the default version to the double precision version (incorrect result from a function--a serious problem). Another problem was an intrinsic function in a certain context returned the wrong answer (a serious problem). Two compile-time errors include: returning a value from an intrinsic function of a defined size when the ANSI standard provides a generic size and requiring the parameters of certain functions to be a specific size when the ANSI standard defines the parameter to be a generic size (two annoyance problems). All of these problems caused the conversion effort to the SGI machines running IRIX 4.x or 5.x to earn a difficult rating.

The machine running IRIX 6.0 had one annoyance problem; an intrinsic function did not operate properly with extended precision. Since this problem was caught by the compiler it was rated an annoyance problem and conversion effort for the SGI running IRIX 6.0 was rated easy.

The conversion effort to the Sun SPARCstations 20/51, 10/51, 10/41, and IPX was easy. A compiler option to use static local variables did not exist; however, the compiler apparently always uses static local variables, so this was not a problem. The Sun version of FORTRAN does not have a timing routine nor a date routine (two annoyance problems). Since the porting effort to the Sun machines only had two annoyance problems, this effort was rated easy.

Conclusions

Before any serious study of computational performance may be undertaken, basic factors like computational accuracy and software porting costs must be verified. This study indicates that, in general, workstation vendors have no problem meeting computational accuracy requirements for the X-29 aircraft simulation model. Moving this simulation model to the different computational platforms shows that there is little middle ground, porting will be either easy or difficult.

Traditionally, only supercomputers have been able to provide the large computational power to operate the X-29 aircraft simulation model in a real-time flight simulation environment; however, modern workstations now provide enough computational power. The X-29 aircraft model was chosen since it is computationally similar to many aircraft simulation models. However, one should be careful not to extrapolate these performance results: by the scalar nature of the benchmark, the vector registers of the CONVEX and Cray computers do not provide the performance enhancement normally associated with vector supercomputers. Some aircraft models, unlike the benchmark used in these tests, may require vector registers (e.g., aircraft modelled with flexible airframes).

Workstations from several manufacturers provide the perquisite computational accuracy and porting costs while suppling sufficient computational performance to operate the X-29 aircraft simulation model in real time.

Table 1 - Machines Tested

Computer	Processor	Speed (MHz)	UNIX OS	OS Version	FORTRAN Version
CONVEX C38xx	Custom vector	-	UXE	1.2	7.0.1.0
CONVEX C32xx	Custom vector	-	UXE	1.2	7.0.1.0
Cray Y-MPE/8	Custom vector	-	Unicos	R6.1	5.0.4.0
Cray-2	Custom vector	-	Unicos	R6.1	5.0.4.0
Digital 300/800	Alpha	200	OSF/1	2.0	3.4
Digital 3000/500X	Alpha	200	OSF/1	-	-
Digital 3000/500	Alpha	150	OSF/1	-	-
HP 9000-735	PA-RISC 7100	99	HP-UX	9.0.1	9.0
HP 9000-725	PA-RISC	50	HP-UX	-	-
IBM PC Clone	i486	66	LynxOS	2.1	2.6.4.5
IBM RS/6000 970	RS/6000	50	AIX	3.2	2.2
IBM RS/6000 560	RS/6000	50	AIX	3.2	2.2
SGI Onyx 8000/75	R8000	75	IRIX	6.0	6.0
SGI Onyx 4400/150	R4400	150	IRIX	5.1.1.2	5.0
SGI Onyx 4400/100	R4400	100	IRIX	5.0	5.0
SGI Crimson	R4000	100	IRIX	4.0.5	3.1
SGI Indy	R4000	100	IRIX	5.1.1	5.0
Sun SPARC 20/50	SuperSPARC	50	SunOS	4.1.3	2.0.1
Sun SPARC 10/51	SuperSPARC	50	SunOS	4.1.3	2.0.1
Sun SPARC 10/41	SuperSPARC	40	SunOS	4.1.3	2.0.1
Sun SPARC IPX	SPARC	40	SunOS	4.1.2	1.4

Table 2 - Benchmark Results

Computer	Time (Sec)	Performance Index	Inaccuracy (percent off)	Portability
HP 9000-735	49	4.90	10.0[1]	Difficult
SGI Onyx 8000/75	51	4.71	0.0001	Easy
Digital 3000 model 800	56	4.29	0.0001	Easy
Cray Y-MP	68	3.53	10.0	Difficult
Digital 3000 model 500X	76	3.16	0.0001	Easy
HP 9000-720	90	2.67	10.0[1]	Difficult
Digital 3000 model 500	101	2.38	0.0001	Easy
SGI Onyx 4400/150	108	2.22	0.0001	Difficult
Cray-2	109	2.20	10.0[1]	Difficult
CONVEX C38xx	117	2.05	0.0001	-
IBM RS/6000 970	120	2.00	0.0001	Moderate
IBM RS/6000 560	140	1.71	0.0001	Moderate
SGI Onyx 4400/100	166	1.45	0.0001	Difficult
SGI Indy	169	1.42	0.0001	Difficult
SGI Crimson	178	1.35	0.0001	Difficult
CONVEX C32xx	240	1.00	0.0001	-
Sun SPARCstation 10/51	253	0.95	0.0001	Easy
Sun SPARCstation 20/50	354	0.68	0.0001	Easy
Sun SPARCstation 10/41	401	0.60	0.0001	Easy
CDC CYBER 175[2]	660	0.36	0.0	-
Sun SPARCstation IPX	1000	0.24	0.0001	Easy
IBM PC Clone 486/66	1201	0.20	0.0001	Difficult

[1]Although results from the Cray-2, the HP 9000-735, and the HP 9000-720 disagreed with the actual results, they agreed with each other. This implies that the benchmark may be sensitive to small differences in one or more computational results.

[2]Data for CDC CYBER 175 was obtained in a previous, unpublished study.