****

Performance of Various Computers Using Standard Linear Equations Software in a Fortran Environment

****

Jack J. Dongarra
Mathematics and Computer Science Divison
Argonne National Laboratory
Argonne, Illinois 60439

In this note we compare a number of different computer sytems for solving dense systems of linear equations using the LINPACK software in a Fortran environment. There are about 50 computers compared, ranging from a Cray X-MP to the 68000 based systems such as the Apollo and SUN workstations.

The timing information presented here should in no way be used to judge the overall performance of a computer system. The results reflect only one problem area: dense systems of equations using the LINPACK [1] programs in a Fortran environment.

The LINPACK programs can be characterized as having a high percentage of floating point arithmetic operations. The routines involved in this timing study, SGEFA and SGESL, use algorithms which are column oriented. By column orientation we mean the programs usually reference array elements sequentially down a column, not across a row. Column orientation is important in increasing efficiency in a Fortran environment because of the way in which arrays are stored. Most of the floating point operations in LINPACK actually take place in a set of subprograms called the Basic Linear Algebra Subprograms (BLAS) [2]. These routines are called by the LINPACK routines repeatedly throughout the calculation. The BLAS reference one-dimensional arrays, rather than two-dimensional arrays.

Note that these numbers are for a problem of order 100. The execution speeds on some machines, particularly the vector computers, may not fully utilize the features of certain machines. (See the appendix for a specific comparision of large scientific computers in Fortran which better reflects their performance.)

The table was compiled over a period of time. Subsequent software and hardware changes to a computer system may affect the timing to some extent.

Anyone interested in adding to or updating this table is encouraged to contact the author whose phone number is (312) 972-7246. His ARPAnet address is DONGARRA@ANL-MCS.

REFERENCES

[1] J. J. Dongarra, J. R. Bunch, C. B. Moler, and G. W. Stewart, LINPACK Users' Guide, SIAM Publications, Phil., PA., 1979.

[2] C. Lawson, R. Hanson, D. Kincaid, and F. Krogh, "Basic Linear Algebra Subprograms for Fortran Usage," ACM Trans. Math. Software, Vol. 5 No. 3, 1979, 306-371.

[3] J. J. Dongarra and S. C. Eisenstat, "Squeezing the Most out of an Algorithm in Cray Fortran," ANL Tech. Memo. ANL/MCS-TM-9, May 1983.

Solving a System of Linear Equations with LINPACK[a] in Full Precision[b]

Computer	Compiler[c]	Ratio[d]	MFLOPS[e]	Time secs	Unit[f] µsecs
Cray X-MP	CFT (Coded BLAS)	.36	33	.021	0.061
CDC Cyber 205	FTN (Coded BLAS)	.48	25	.027	0.079
Cray X-MP	CFT	.57	21	.032	0.093
Cray-1S	CFT (Coded BLAS)	.68	18	.038	0.11
Cray-1S	CFT	1	12	.056	0.16
CDC Cyber 205	FTN	1.5	8.4	.082	0.24
CDC 7600	FTN (Coded BLAS)	2.6	4.6	.148	0.43
CDC Cyber 176	FTN 5.1 opt=2	2.6	4.6	.148	0.43
Amdahl 5860	H enhan opt=3 HSFPF	3.1	3.9	.176	0.51
Amdahl 5860	VS opt=3 HSFPF	3.2	3.8	.181	0.53
CDC 7600	FTN	3.8	3.3	.210	0.61
FPS-164	D,opt=3 (Coded BLAS)	4.7	2.6	.264	0.77
IBM 370/195	H enhanced opt=3	4.9	2.5	.275	0.80
IBM 3081 K	H enhanced opt=3	5.7	2.1	.321	0.94
IBM 3081 K	VS opt=3	6.2	2.0	.348	1.01
CDC 7600	Local	6.4	2.0	.359	1.05
IBM 3033	H enhanced opt=3	7.0	1.8	.390	1.14
IBM 3033	VS opt=3	7.1	1.7	.396	1.15
Amdahl 470 V/8	H enhanced opt=3	7.7	1.6	.429	1.25
Amdahl 470 V/8	VS opt=3	8.2	1.5	.458	1.33
CDC Cyber 175	FTN 4.6 opt=2	8.7	1.4	.489	1.42
CDC Cyber 175	FTN ext 4.6 opt=1	9.0	1.4	.508	1.47

Computer	Compiler[c]	Ratio[d]	MFLOPS[e]	Time secs	Unit[f] µsecs
FPS-164	D,opt=3	9.5	1.3	.529	1.54
CDC 7600	CHAT, No opt	9.9	1.2	.554	1.61
IBM 370/168	H Ext Fast Mult	10	1.2	.579	1.69
Amdahl 470 V/6	H opt=2	11	1.1	.631	1.84
IBM 370/165	H Ext Fast Mult	16	.77	.890	2.59
ELXSI	EMBOS, F77 (Coded BLAS)	22	.56	1.23	3.57
CDC 6600	FTN 4.6 opt=2	26	.48	1.44	4.19
ELXSI	EMBOS, F77	28	.43	1.60	4.66
UNIVAC 1100/81	ASCII opt=ZEO	32	.38	1.80	5.24
CDC 6600	RUN	34	.36	1.93	5.62
Data General MV/10000	f77 opt level 2	40	.30	2.26	6.58
Harris 800	Fortran 77	53	.23	2.99	8.70
IBM 370/158	H opt=3	53	.23	2.99	8.71
IBM 370/158	VS opt=3	56	.22	3.15	9.17
Itel AS/5 mod 3	H	63	.19	3.54	10.3
IBM 4341 MG10	VS opt=3	66	.19	3.70	10.8
VAX 11/780 FPA	VMS (Coded BLAS)	76	.16	4.25	12.4
ICL 2988	f77 OPT=2	85	.14	4.78	13.9
VAX 11/750 FPA	VMS (Coded BLAS)	88	.14	4.92	14.3
VAX 11/780 FPA	VMS	98	.13	5.48	16.0
Ridge 32	Fort 77	100	.12	5.61	16.3
CDC 6500	FUN	102	.12	5.69	16.6
Denelcor HEP	f77	107	.11	5.98	17.4
VAX 11/780 FPA	UNIX xf77	107	.11	5.98	17.4
VAX 11/750 FPA	VMS	119	.10	6.66	19.4
Prime 850	Primos	130	.095	7.26	21.1
UNIVAC 1100/62	ASCII opt=ZEO	132	.093	7.38	21.5
Data General MV/8000	f77 opt level 2	157	.078	8.80	25.6
VAX 11/750	VMS	216	.057	12.1	35.3
HP 9000	Fortran 1.7	285	.043	16.0	46.6
VAX 11/730 FPA	VMS (Coded BLAS)	286	.043	16.0	46.6
VAX 11/725 FPA	VMS (Coded BLAS)	286	.043	16.0	46.6
Apollo	4.1 PEB (Coded BLAS)	323	.038	18.1	52.7
IBM 4331	H opt=3	326	.032	21.6	62.9
VAX 11/730 FPA	VMS	348	.036	19.5	56.9
VAX 11/725 FPA	VMS	348	.036	19.5	56.9
Prime 2250	Fortran 77	365	.034	20.5	59.6
IBM PC-XT/370	H opt=3	391	.031	21.9	63.7
SUN 2 + SKY board	UNIX, f77 opt	557	.022	31.2	90.1
Apollo	4.1 PEB	559	.022	31.3	91.2
Canaan	VS	588	.021	33.0	96.0
SUN 2 + SKY board	UNIX, f77 no opt	687	.018	38.5	112.
Masscomp MSW500	UNIX, f77	1316	.0093	73.7	215.
SUN 2	UNIX, f77 no opt	2107	.0058	118.	344.
SUN	UNIX, f77 no opt	2661	.0046	149.	434.
Masscomp MC500	UNIX, f77	2661	.0046	149.	434.

Comments:

The Cray X-MP timings reflect only one processor.

The Denelcor HEP run was on a one PEM machine with no parallel constructions. An order of magnitude speedup may be achieved by using the parallel features.

Solving a System of Linear Equations with LINPACK[a] in Half Precision[b]

Computer	Compiler[c]	Ratio[d]	MFLOPS[e]	Time secs	Unit[f] µsecs
Amdahl 5860	H enhan opt=3 HSFPF	2.2	5.5	.125	0.36
Amdahl 5860	VS opt=3 HSFPF	2.4	5.1	.135	0.39
Amdahl 470 V/8	H enhanced opt=3	4.4	2.8	.246	0.71
Amdahl 470 V/8	VS opt=3	4.5	2.7	.254	0.74
IBM 3081 K	H enhanced opt=3	5.1	2.4	.283	.82
IBM 3081 K	VS opt=3	5.6	2.2	.311	.91
IBM 3033	VS Fortran	6.3	1.9	.353	1.03
ELXSI	EMBOS, F77 (Coded BLAS)	17	.71	.967	2.82
ELXSI	EMBOS, F77	23	.51	1.35	3.92
UNIVAC 1100/81	ASCII opt=ZEO	24	.52	1.32	3.85

Computer	Compiler[c]	Ratio[d]	MFLOPS[e]	Time secs	Unit[f] µsecs
Data General MV/10000	f77 opt level 2	31	.39	1.75	5.09
VAX 11/780 FPA	VMS (Coded BLAS)	37	.33	2.08	6.07
Ridge 32	Fort 77 (Coded BLAS)	39	.31	2.19	6.38
IBM 370/158	H opt=3	42	.29	2.35	6.86
DEC KL-20	F20	46	.27	2.59	7.53
IBM 370/158	VS opt=3	46	.26	2.60	7.58
UNIVAC 1100/82	ASCII opt=ZEO	49	.25	2.77	8.09
ICL 2988	f77 OPT=2	50	.25	2.79	8.13
Harris 800	Fortran 77	53	.23	2.99	8.70
VAX 11/750 FPA	VMS (Coded BLAS)	56	.22	3.14	9.16
IBM 4341 MG10	VS opt=3	57	.22	3.18	9.25
VAX 11/780 FPA	VMS	59	.21	3.28	9.57
VAX 11/780 FPA	UNIX xf77	61	.20	3.41	9.93
Honeywell 6080	Y	62	.20	3.46	10.1
Ridge 32	Fort 77	62	.20	3.48	10.1
Data General MV/8000	f77 opt level 2	69	.18	3.84	11.2
VAX 11/780	VMS	74	.17	4.13	12.0
VAX 11/750 FPA	VMS	86	.14	4.80	14.0
Prime 850	Primos	97	.13	5.41	15.8
HP 9000	Fortran 1.7	125	.098	7.00	20.4
VAX 11/750	VMS	137	.089	7.69	22.4
IBM 4331	H opt=3	140	.088	7.84	22.8
Apollo	4.1 PEB (Coded BLAS)	177	.069	9.92	28.9
VAX 11/730 FPA	VMS (Coded BLAS)	205	.060	11.5	33.4
VAX 11/725 FPA	VMS (Coded BLAS)	205	.060	11.5	33.4
Burroughs 6700	H	234	.052	13.1	38.2
Prime 2250	Fortran 77	258	.048	14.5	42.1
VAX 11/730 FPA	VMS	259	.047	14.5	42.2
VAX 11/725 FPA	VMS	259	.047	14.5	42.2
IBM PC-XT/370	H opt=3	303	.040	17.0	49.5
DEC KA-10	F40	305	.040	17.1	49.8
Canaan	VS	306	.040	17.1	49.9
SUN 2 + SKY board	UNIX, f77 opt	314	.039	17.6	51.1
Apollo	4.1 PEB	334	.037	18.7	54.5
SUN 2 + SKY board	UNIX, f77 no opt	430	.029	24.1	43.0
Masscomp MVS500	UNIX, f77	873	.014	48.9	142.
IBM PC/8087	Microsoft 3.1	1071	.011	60.0	175.
SUN 2	UNIX, f77 no opt	1029	.012	57.6	168.
Masscomp MC500	UNIX, f77	1245	.0099	69.7	203.
SUN	UNIX, f77 no opt	1298	.0094	72.7	212.
IBM PC	Microsoft 3.1	21875	.00056	1225.	3568.
Apple III	Pascal	50232	.00024	2813.	8193.

[a] LINPACK routines SGEFA and SGESL were used for single precision and routines DGEFA and DGESL were used for double precision. These routines perform standard LU decomposition with partial pivoting and backsubstitution.

[b] "Full Precision" implies the use of (approximately) 64-bit arithmetic, e.g. CDC single precision or IBM double precision. "Half Precision" implies the use of (approximately) 32-bit arithmetic, e.g. IBM single precision.

[c] "Compiler" refers to the compiler use and "(Coded BLAS)" refers to the use of assembly language coding of the BLAS.

[d] "Ratio" is the number of times faster or slower a particular machine configuration is when compared to the Cray-1S using a Fortran coding for the BLAS.

[e] "MFLOPS" is a rate of execution, the number of million floating point operations completed per second. For solving a system of equations there are approximately $2/3n^3 + 2n^2$ operations performed (we count both additions and multiplications).

[f] "Unit" is the time in microseconds required to execute the statement $y_i = y_i + t*x_i$. This involves one floating point multiplication, one floating point addition, and a few one-dimensional indexing operations and storage references. The actual statement occurs in SAXPY, which is called roughly n^2 times by SGEFA and n times by SGESL with vectors of varying lengths. The statement is executed approximately $1/3\ n^3 + n^2$ times. Thus for n = 100,

$$\text{Unit} = 10^6 \text{Time}/[(1/3)*100^3 + 100^2].$$

Performance of Large Scientific Computers in a Fortran Environment

The LINPACK routines used to generate the timings in the previous table do not reflect the true performance of "advanced scientific computers." A different implementation of the solution of linear equations, presented in a report by Dongarra and Eisenstat[3], better describes the performance on such machines. That algorithm is based on matrix-vector operations rather than just vector operations. This produces a program that has a high level of modularity or larger granularity, having the potential for better performance across a wide range of machines, especially on high performance computers. As before, a Fortran program was run and the time to complete the solution of equations for a matrix of order 300 is reported.

Note that these numbers are for a problem of order 300 and all runs are for full precision.

The table was compiled over a period of time. Subsequent software and hardware changes to a computer system may affect the timing to some extent.

Solving a System of Linear Equations Using the Vector Unrolling Technique

Computer	Compiler[a]	MFLOPS[b]	Time secs	Unit[c] μsecs
Cray X-MP ‡	CFT (Coded ISAMAX)	240	.076	.0083
Cray X-MP ‡	CFT	161	.113	.012
Cray X-MP †	CFT (Coded ISAMAX)	134	.136	.015
Cray X-MP †	CFT	106	.172	.019
Cray 1-M	CFT (Coded ISAMAX)	83	.215	.024
Cray 1-S	CFT (Coded ISAMAX)	76	.236	.026
Cray 1-M	CFT	69	.259	.029
Cray 1-S	CFT	66	.273	.030
IBM 370/195	VS opt=2	4.4	4.1	.455
FPS 164	D,opt=3 (Coded ISAMAX)	4.1	4.4	.488
FPS 164	D,opt=3	4.0	4.5	.500
IBM 3033	VS opt=2	2.5	7.1	.800
VAX 11/780 FPA	UNIX xf77	.11	177.	19.5

Comments:

‡ The Cray X-MP timings reflect the use of two processor in solving the problem.

† The Cray X-MP timings reflect only one processor.

a "Compiler" refers to the compiler used and "(Coded ISAMAX)" refers to the use of assembly language coding of the BLAS ISAMAX.

b "MFLOPS" is a rate of execution, the number of million floating point operations completed per second. For solving a system of equations there are $2/3n^3 + 2n^2$ operations performed (we count both additons and multiplications).

c "Unit" is the time in microseconds required to execute the statement $y_i = y_i + t*x_i$. This involves one floating point multiplication, one floating point addtion, and a few one-dimensional indexing operations and storage references.

The major difference between the Cray 1-M and Cray 1-S is in the memory speed, the Cray 1-M has slower memory. The timings show the Cray 1-M to be faster than the Cray 1-S. After much discussion and examination of the generated assembly language code it was determined that, in fact, the Cray 1-M was faster for this program. The code generated by the compiler causes the Cray 1-S to miss a chain-slot. On the Cray 1-M, because of slower memory, the chain-slot is not missed, thus the faster execution time.