On the efficiency of superscalar and vector computer for some problems in scientific computing*

M. Tůma and M. Rozložník

Institute of Computer Science, Academy of Sciences of the Czech Republic
Pod vodárenskou věží 2, 182 07 Praha 8, Czech Republic
e-mail: tuma@uivt.cas.cz, miro@uivt.cas.cz

Abstract. Some details of arithmetic of two representatives of computers (a superscalar workstation and a vector uniprocessor) available in the Czech Republic for scientific computing are described. Consequently, their efficiency and precision on a set of linear algebraic tasks solved by different solvers is compared.

1 Introduction

One very important question in scientific computing is how to guarantee efficient performance of a given computer on a set of fundamental operations. On the other hand, efficiency on a set of basic, often linear algebra operations, plays usually the key role in the choice of computer for scientific computations.

Although a lot of effort has been devoted to developing benchmarks for scientific computing [5], [9], [2] there is still a lack of comparison for computers avilable using a broader spectrum of the important basic algorithms.

We devote our attention to some properties of computer arithmetic of two computers which present different philosophy of performing arithmetic operations. Their choice was motivated by the hardware platforms available for the scientific computations in the Czech Republic: computer Cray Y-MP EL installed in the Institute of Physics of the Czech Academy of Sciences, the representative of vector processing (we concentrate on the vector parallelism only), SGI Crimson with the superscalar processor R4000, the representative of the group of advanced uniprocessor workstations installed at many places in the Czech Republic. The comparison of their efficiency and precision is based on the set of numerical experiments. Our investigation includes not only CPU timings, but also precision of solutions and residuals.

We concentrate on the one of the fundamental problems of the computational linear algebra: solving large linear algebraic systems with symmetric and positive definite, indefinite or nonsymmetric system matrix. This task is crucial for successful tackling of many physical, chemical and economical applications. For

* This work was supported in part by the Grant Agency of the Czech Republic with grant No. 201/93/0067.

instance, core of both interior point and simplex linear programming methods is completely dependent on the sparse linear solvers. The most dominant part of the CPU time spent for the numerical solution of partial differential equations describing various phenomena in physics and chemistry is usually spent in linear solver of the discretized equation.

Efficient solution of large linear systems needs a careful choice of the algorithm with respect to the problem size, problem characteristics and the computer architecture available. Therefore, we use different algorithmic techniques to solve the same problem.

2 Precision of numerical computations

Let Ψ denote the **machine precision** defined as $1^+ - 1$, where 1^+ is the successor of 1 in the machine arithmetic. We set $\Psi = \beta^{-p}$. Superscalar workstation SGI Crimson performs the arithmetic calculations according to IEEE standard, where $\beta = 2$ and $p = 53$. The arithmetic of vector machine Cray Y-MP EL satisfies $\beta = 2$ and $p = 50$. Let us define the roundoff unit $\mathbf{u}$ as the highest number which does not change 1 if added to it. This roundoff unit depends on the machine precision and the way in which is the result rounded. The workstations use rounding to the nearest, which results in $\mathbf{u} = \frac{1}{2}\Psi$. The Cray Y-MP EL uses rounding with chopping which implies $\mathbf{u} = \Psi$. These machine constants and the computer clock frequencies are tabulated in Table 1.

Values	Cray Y-MP EL	SGI Crimson
Ψ	1.8D-15	2.2D-16
$\mathbf{u}$	1.8D-15	1.1D-16
Clock frequency	33.3 MHz	50 MHz

Table 1: Comparison of the machine precision, roundoff unit and clock frequency for the computers Cray Y-MP EL and SGI Crimson

The difference in the precision of the basic hardware of the two computers can be easily faced in practice. Using Cray Y-MP EL for the computations, we often need to use an additional procedure to make the results of numerical algorithms more precise, when the problem dimension is reasonably large. J. Demmel in [4] quotes a result of the linear system solver where the iterative solution improvement had to be used to get acceptable solution. The differences in precision of the computed results are clearly seen from our experiments in subsequent sections.

The basic benchmarks for the solution of generated dense linear systems are being published in the regularly updated report [5]. We show the performance results for our computers for the test problems of dimensions 100 and 1000 (denoted as L100 and L1000) in Table 2. In addition, we give the theoretical peak performance corresponding to the CPU results with the problem which is either totally vectorized (for Cray Y-MP) and for which the superscalar processor reaches its maximum speed for the whole problem. Details about the compiler options can be found in [5]. We present the results in the millions of floating-point operations per second (MFLOPS).

Performance	Cray Y-MP EL	SGI Crimson
L100	32 MFLOPs	16 MFLOPs
L1000	107 MFLOPs	32 MFLOPs
Peak performance	133 MFLOPs	50 MFLOPs

Table 2: Comparison of computer performance for the dense linear systems computations

Benchmarks based on the dense linear system solvers are easy to construct, but they often say very little about the application problems. The problems which we use for the tests are often much closer to the reality. Nevertheless, they also cannot be exhausting, as well as any isolated idea used in computer benchmarks construction (cf. [3], [9]).

Probably the biggest advantage of the Cray Y-MP EL over an advanced workstation is its vector processor, which enables to manipulate the vectors of data in an effective way. This is advantageous especially in dense matrix manipulations, where operations with dense vectors and submatrices constitute prevailing part of the overall operations. Moreover, the processing of sparse vectors and matrices (vectors and matrices with many zero elements taking into account only the nonzeros), which have elements indirectly addressed, is also supported by hardware. The Cray computer uses so-called hardware gather-scatter enabling to vectorize also the sparse calculations.

The workstation SGI Crimson does not have an vector processing unit. Nevertheless, its superscalar RISC unit is very efficient. It enables to pipeline one scalar addition with one scalar multiplication in the so-called superscalar mode. Moreover, its integer arithmetic unit is more efficient than the Cray's one.

All the programs used were written in Fortran 77 using double precision. Some of the solvers were written by the authors, including the sparse matrix codes. Other were taken from the Linpack and Lapack (see [1], [6]). For timings we used the standard function *etime()*, programs were optimized using the highest compiler options. Test matrices were taken from the Harwell-Boeing collection. We will shortly quote the application areas to give the idea about the scope of the comparison: linear programming, economical modelling, air-pollution including chemical kinetics and transport equations, computer simulations, electrical engineering problems, structural engineering, power network calculations, chemical engineering, astrophysics, nuclear reactor computations and flow modelling. The right-hand side vectors were computed using the known solution with unit components.

3 Linear system solvers

3.1 Symmetric positive definite case

Consider the system of linear algebraic equations

$$Ax = b,$$

where $A \in R^{N,N}$ is a symmetric and positive definite (SPD) matrix and $b \in R^N$ is a right-hand side vector. Standard strategy for the SPD computations is the symmetric form of the Gaussian elimination (For discussion on the iterative and direct methods, sparsity and computer architectures see [10]). Among the most effective implementations of the dense algorithms are the ones in Linpack ([6]) and Lapack ([1]) libraries. We used the procedures DPOFA and DPOSL from Linpack library. The algorithms are usually incorporated in basic libraries delivered by computer producers. The results given in the Table 3 clearly show that in the dense case, the procedures can be easily vectorized and the CPU times obtained from the Cray Y-MP EL are better that the CPU times from the workstation, especially for larger problems.

	Dimension	Nonzeros	CPU time		Error norm		Residual norm	
MATRIX	N	NONZ	Cray	Crimson	Cray	Crimson	Cray	Crimson
1138BUS	1138	2596	27.88	66.29	1.05–06	7.60–10	2.85–08	4.61–11
494BUS	494	1080	4.44	4.75	1.25–08	2.16–11	1.42–09	1.29–11
662BUS	662	1568	8.36	11.7	8.51–09	3.89–12	1.62–10	9.75–13
685BUS	685	1967	9.00	13.07	1.50–09	3.85–13	3.11–10	8.29–12
BRNO974	974	12340	19.61	41.03	1.14–09	1.52–12	2.93–08	1.49–10
GR3030	900	4322	16.45	31.61	6.35–11	2.94–14	5.44–12	4.92–14

Table 3: Results of comparison for the dense implementation of the symmetric Gaussian elimination of SPD matrices

For most of the matrices, the nonzero elements constitute only a small proportion of all the matrix elements. Nowadays, an enormous effort is devoted to sparse implementations of the computer algorithms, which enable to make use of the structure of nonzeros in the original matrix. In SPD case we can determine the structure of nonzeros in the matrix obtained from the symmetric Gaussian elimination in advance. This leads to efficient implementations which make use of the indirectly addressed nonzero elements processed during the algorithm. For the sake of the comparison we used our implementation which does not promote any special advantages of Cray or workstation based on the column decomposition (see [7]). It is out of scope of this paper to compare various enhancements (e.g. block building) of the basic algorithmic schemes and their influence on the code performance (see [8]). The results are provided in Table 4.

	Dimension	Nonzeros	CPU time		Error norm		Residual norm	
MATRIX	N	NONZ	Cray	Crimson	Cray	Crimson	Cray	Crimson
1138BUS	1138	2596	0.42	0.07	0.23–09	0.65–12	0.19–05	0.66–08
494BUS	494	1080	0.16	0.03	0.40–10	0.62–12	0.17–09	0.18–11
662BUS	662	1568	0.30	0.05	0.18–09	0.14–13	0.89–10	0.23–12
685BUS	685	1967	0.34	0.05	0.25–10	0.94–13	0.12–09	0.36–11
BRNO974	974	12340	2.30	0.57	0.12–08	0.59–12	0.28–07	0.56–10
GR3030	900	4322	0.79	0.15	0.74–11	0.33–14	0.21–11	0.53–14

Table 4: Results of comparison for the sparse implementation of the symmetric Gaussian elimination of SPD matrices

Our results show that the timings for the workstation are invariably better. For the discussion what are the algorithmic features promoting this fact see [10]. Moreover, all the results are better for the sparse algorithm on all the test matrices. We face the problem of reasonable benchmarks again, since the dense linear solver test is often not sufficient to reveal the real computer performance.

As for the precision (norm of the error of the solution $\hat{x}-x$) and residual size (norm of $b-A\hat{x}$), where $\hat{x}$ is the computed approximation to the solution vector, we can notice that they differ as the theory predicts. In particular, results from Cray Y-MP EL are usually less precise and iterative improvement is sometimes necessary for larger and/or ill-conditioned problems.

3.2 Systems with symmetric indefinite and nonsymmetric matrices

The typical source of these problems are bases of the linear programming problems, economical and chemical engineering and discretized flow problems. Table 5 provides results of the direct sparse nonsymmetric solver for the symmetric indefinite and nonsymmetric matrices.

	Dimension	Nonzeros	CPU time		Error norm		Residual norm	
MATRIX	N	NONZ	Cray	Crimson	Cray	Crimson	Cray	Crimson
BCSSTK09	1083	9760	34.91	7.71	0.91–08	0.75–11	0.13–03	0.37–06
BP800	822	4534	2.48	0.45	0.17–09	0.29–11	0.14–09	0.34–12
FS5413	541	4285	2.92	0.59	0.79–06	0.80–09	0.94–06	0.12–08
GRE1107	1107	5664	13.53	3.47	0.17–06	0.29–09	0.19–11	0.58–13
JPWH991	991	6027	11.94	2.65	0.18–09	0.19–13	0.44–10	0.62–13
MAHINDAS	1258	7682	5.72	1.21	0.24–07	0.45–09	0.58–07	0.74–09
NNC666	666	4044	21.05	9.02	0.75–04	0.33–06	0.10–09	0.21–11
NOS7	729	2673	9.71	2.13	0.22–03	0.50–08	0.63–05	0.66–08
ORSIRR1	1030	6858	9.21	1.97	0.51–08	0.41–11	0.19–06	0.34–09
PORES2	1224	9613	33.84	9.49	0.10–07	0.15–11	0.32–05	0.21–07
SHERMAN1	1000	3750	2.47	0.51	0.23–09	0.75–12	0.10–11	0.58–14
WEST0989	989	3537	4.09	0.57	0.26–07	0.35–09	0.73–08	0.21–09

Table 5: Results of comparison for the sparse implementation of the Gaussian elimination for symmetric indefinite and nonsymmetric matrices

Also as in the SPD case, the results are in favor of the superscalar computer. The partial reason is the increased portion of the integer calculations due to pivoting. Moreover, the timings are strongly influenced by the structure of nonzeros of the matrix. For the Cray Y-MP EL computer, the dense implementation provided better timings for some small problems. This never happened for the SGI computer.

4 Conclusions

The paper contains a comparison of the two representatives of computers available for scientific computations in the Czech Republic. We concentrated on the precision, issue of density and sparsity in computer benchmarks, fitting the algorithm to the architecture for the fundamental linear algebra problem arising in important applications in various branches of industry.

Although algorithms with dense matrices are very often used in the benchmarks and they promote a good speedup on the vector computers, the real-world problems provide rather matrices with many nonzero elements - sparse matrices. Processing of dense blocks is only a subtask of a complicated direct sparse solver. The necessary indirect addressing decreases the potential for the vectorization. In this case, the superscalar architecture is often preferable. The results are then often not in favor for the Cray Y-MP EL computer and the proportion price/performance is much better for the superscalar workstation. We also note the problems with precision for the Cray Y-MP computers.

In any case, extrapolation of the results to other members of the Cray Y-MP and SGI families must be done very carefully and detailed discussion is outside the scope of this paper. We have compared the computers which are available for the computations in the Czech Republic, demonstrated some of their characteristics and pointed out that the choice of a computer for scientific computing is a complicated multidimensional problem.

References

1. Anderson, E., Bai, Z., Bischof, J., Demmel, J., Dongarra, J.J., Du Croz, J., Greenbaum, A., Hammarling, S., McKenney, A., Ostrouchov, S., Sorensen, D.: Lapack user's guide. SIAM Philadelphia, 1992
2. Berry, M., Cybenko, G., Larson, J.: Scientific benchmarks characterizations. Parallel Computing **17** (1991), 1173–1194.
3. Daydé, M.J., Duff, I.S.: Porting industrial codes and developing sparse linear solvers on parallel computers, Technical Report RAL 94-019, Rutheford Appleton Laboratory, 1994.
4. Demmel, J.W.: Trading off parallelism and numerical stability. In: Linear Algebra for Large-Scale and Real-Time Applications (M.S.Moonen et al. eds.), Kluwer Academic Publishers 1993, 49-68
5. Dongarra, J.J.: Performance of various computers using standard linear equations software. Technical Report CS-89-85, University of Tennessee, Update 1.2.1995
6. Dongarra, J.J., Bunch, J.R., Moler, C.B., Stewart, G.W.: Linpack user's Guide. SIAM, Philadelphia 1979
7. Liu, J.W.H.: The role of elimination trees in sparse factorization. SIAM J. Matrix Anal. Appl. **11** (1990), 134–172
8. Ng, E.G., Peyton, B.W.: Block sparse Cholesky algorithms on advanced uniprocessor architectures, SIAM J. Sci. Comput. **14** (1993), 1034–1056.
9. Pointer,L: Perfect: Performance evaluation for cost-effective tansformations. Report 2. CSRD Report No. 964, University of Illinois 1990
10. Strakoš, Z., Tůma, M.: Current trends in numerical linear algebra: from theory to practice. In: Proceedings of SOFSEM'94, 1994, 229–247