Parallel Matrix Multiplication -- Can we Learn Anything New?

Marcin Paprzycki, Dept. of Math & Computer Science, UT Permian Basin

This note is a follow-up to the results presented in [4] where we examined matrix multiplication on one processor of the Cray Y-MP. It was shown that when the level 3 BLAS [3] routine SGEMM [1] is used, the practical peak performance can be estimated at approximately 315 MFlops. It was also demonstrated that the Strassen's algorithm-based routine SGEMMS [1,5] multiplies matrices faster than SGEMM (for matrices of size bigger than 200). The cost of the additional speed is extra memory required.

In September 1991, the CHPC upgraded the Cray operating system to UNICOS 6.1. The new version of the system provides parallel (level 1, 2, and 3) BLAS kernels and more support for algorithm parallelization in general. Recently, we have learned that since the beginning of the school year 1991-92, there is an increased interest in exploiting the Cray's parallelism. We would therefore like to discuss some experimental results offering insight into parallelization on the Cray. Our experiments were, as previously, performed with matrix multiplication that parallelizes almost perfectly and can thus be a useful guideline for performance expectations.

Our experiments were executed on two Cray Y-MP computers. One of them belongs to CHPC, while the second is a part of Cray's research facility in Eagan, Minnesota. Both machines were running UNICOS 6.1 and generated analogous results. Each result presented here is an average of multiple runs on both machines. Since the *perftrace* utility does not work in a multiple processor environment (this problem will be, most likely, solved by UNICOS 7.0), all the jobs were run either in a single job stream environment ("benchmark" queue on charon) or in a dedicated mode with nobody else using the system.

In the first series of experiments, we tried to establish the minimum matrix size that will parallelize efficiently on all 8 processors. We utilized the Assembly-coded level 3 BLAS routine SGEMM. Speedups obtained for matrix sizes varying from 100 to 500 are presented in Figure 1.

It is only when matrix size is larger than 400 that an almost linear speedup on eight processors is observed. When, on the other hand, up to four processors are used ("parallel" queue on charon), it is enough for matrix size to exceed 300 to achieve linear speedup. A standard matrix multiplication process, for a matrix of size N, requires N**3 multiplications and N**3 – N**2 additions, which adds up to 2N**3 – N**2 floating point operations.

Therefore, it can be calculated that for N > 400 the eight processor system delivers about 2400 MFLOPs (about 300 MFLOPs per processor). For large matrices (N > 1500), the eight processor MFLOP rate increases to about 2490. In this case, each processor delivers about 310 MFLOPs.

The practical peak performance (about 315 MFLOPs) can thus be regarded as virtually attained, since some overhead needs to be taken into account as well. It is thus fair to say that 2500 MFLOPs is a practical peak performance of the eight-processor Cray Y-MP.

Our second goal was to compare the parallel efficiency of the standard matrix multiplication routine SGEMM)with one based on Strassen's algorithm, SGEMMS. The ratio of SGEMMS time to SGEMM time was used to compare the performance. Figure 2 presents the results for 1, 2, 4, 6, and 8 processors. The results for 3, 5, and 7 processors were omitted for clarity. Their curves, however,

fall exactly between the curves representing the performance of 2 and 4, 4 and 6, 6 and 8 processors, respectively.

The results were quite surprising. In all cases, sudden drops in the performance of SGEMMS were observed. It was only for a one processor system that SGEMMS proved more efficient than SGEMM in the whole range of matrix sizes. But even in this case, the same pattern of performance drops is observed. In order to be sure whether the observed phenomenon has nothing to do with memory bank conflicts, we decided to locate the critical matrix sizes. Figure 3 presents the results for N between 1030 and 1044. The performance was again presented as a ratio of the SGEMMS time to the SGEMM time.

The observed phenomenon was explained by Michael Merchant from Cray Research, the implementor of SGEMMS. SGEMMS is a recursive procedure programmed in Fortran where the block operations are performed by calls to SGEMM. What we observe here is caused by recursive divisions going all the way down to blocks of size 65. This is a "very bad" vector length (64 + 1) that does not allow for effective use of the vector processor, slowing down calculations considerably.

This effect explains the overall shape of the curves on Figure 2, since the turning points can be located at matrix sizes 260, 520, 1040, and 2080. The division all the way to blocks of size 65 allowed us also to observe a trade-off between the efficiency of parallel operations on small matrices and gains from the recursive nature of the Strassen's method. Theoretically, performance should gain with every level of recursion, but small block sizes will nullify the gains for a larger number of processors.

According to Michael Merchant, this problem has been fixed by reducing the level of recursion used in SGEMMS by one (the smallest block will be of size 130). This fix will be included UNICOS 7.0.

Summarizing, we can learn something from simple matrix multiplication. Our results suggest that in order to expect good speedups, the problem must be at least as intensive computationally as matrix multiplication of matrices of size 400. If the problem does not parallelize as nicely as matrix multiplication, or is not coded in the Cray Assembly Language, one must make it appropriately bigger. For the time being, we recommend that you avoid using Strassen's algorithm-based matrix multiplication for multiprocessor applications not only because of stability problems [3], but also because of possible inefficiencies.

References

1. Cray Research, Inc., *Math and Scientific Reference Manual*, SR-2081 5.0.
2. Dongarra, J. J., Du Croz, J., Duff, I., Hammarling, S., "A Set of Level 3 Basic Linear Algebra Subprograms," Technical Report ANL-MCS-TM57, Argonne National Laboratory, 1988.
3. Higham, N. J., "Exploiting Fast Matrix Multiplication Within the Level 3 BLAS," Technical Report TR 89-984, Cornell University, 1989.
4. Paprzycki, M., Cyphers, C., "Multiplying Matrices on the Cray -- Practical Considerations," CHPC Newsletter, 6 (6), 1991, 77-82.
5. Strassen, V., "Gaussian Elimination is not Optimal," Numerical Mathematics, 13, 1969, 354-356.

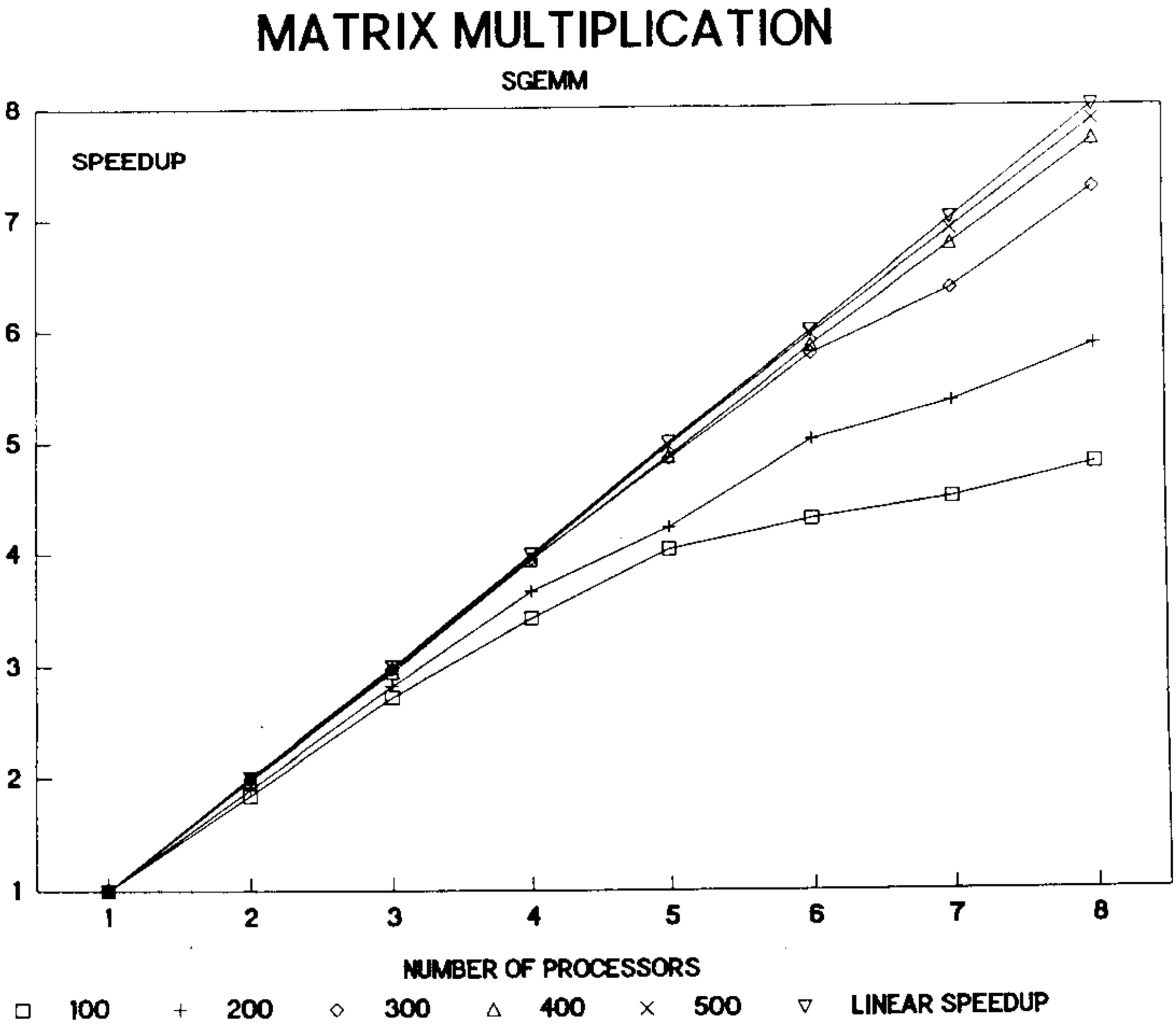
MATRIX MULTIPLICATION
SGEMM
SPEEDUP
8
7
6
5
4
3
2
1
1
2
3
4
5
6
7
8
NUMBER OF PROCESSORS
□ 100 + 200 ◇ 300 △ 400 × 500 ▽ LINEAR SPEEDUP

Figure 1.

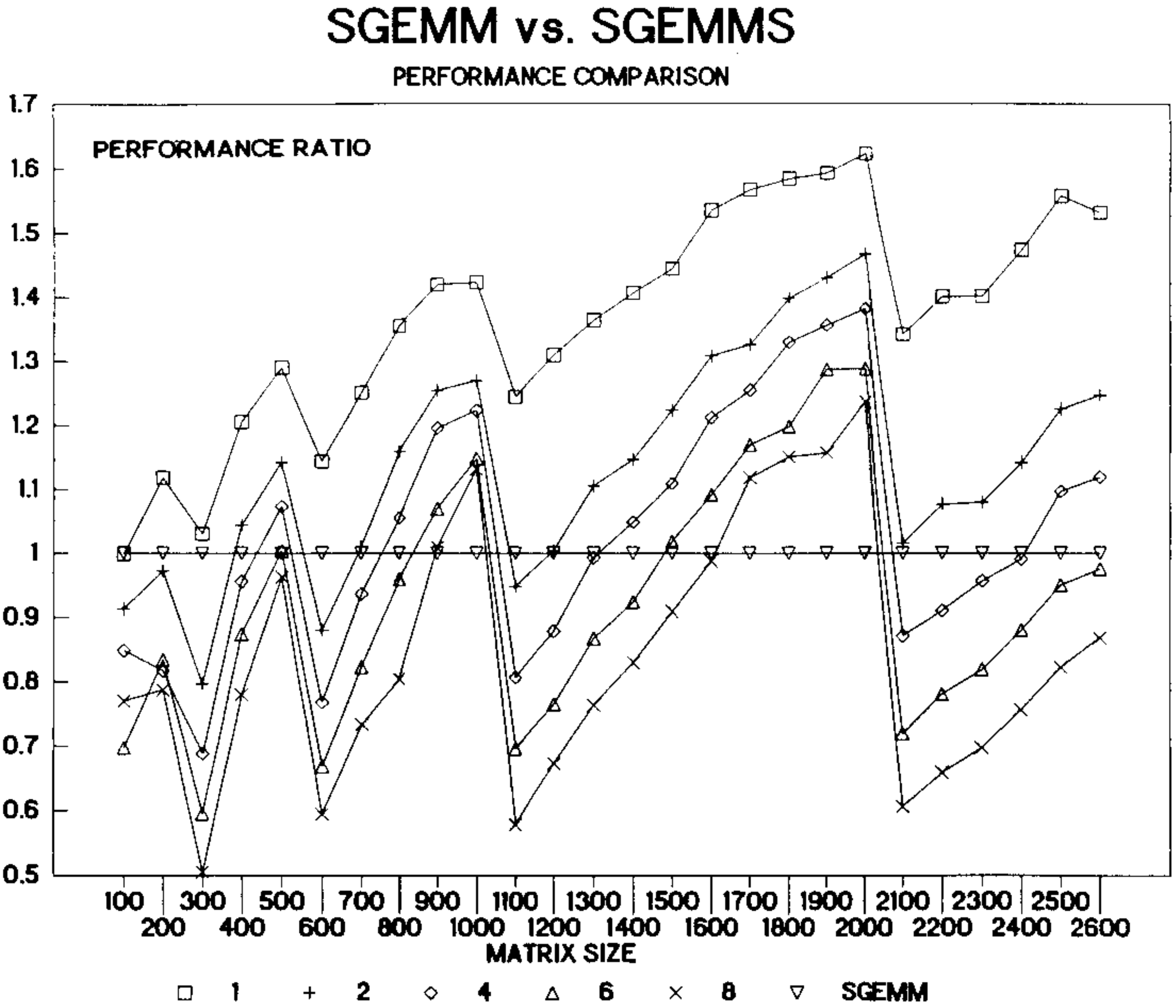
SGEMM vs. SGEMMS
PERFORMANCE COMPARISON
PERFORMANCE RATIO
1.7
1.6
1.5
1.4
1.3
1.2
1.1
1
0.9
0.8
0.7
0.6
0.5
100 300 500 700 900 1100 1300 1500 1700 1900 2100 2300 2500
200 400 600 800 1000 1200 1400 1600 1800 2000 2200 2400 2600
MATRIX SIZE
□ 1 + 2 ◇ 4 △ 6 × 8 ▽ SGEMM

Figure 2.

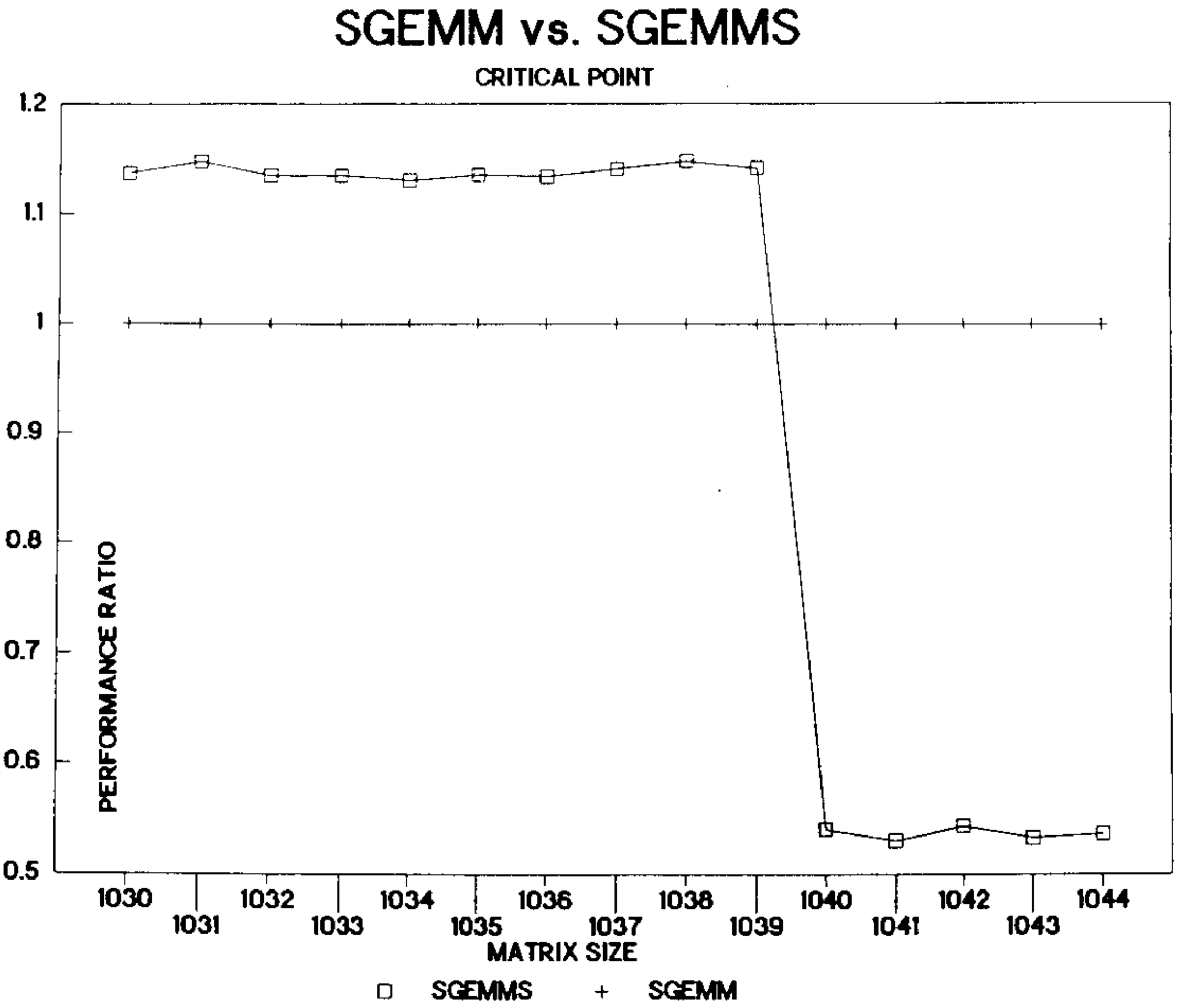
SGEMM vs. SGEMMS
CRITICAL POINT
1.2
1.1
1
0.9
0.8
0.7
0.6
0.5
PERFORMANCE RATIO
1030
1031
1032
1033
1034
1035
1036
1037
1038
1039
1040
1041
1042
1043
1044
MATRIX SIZE
SGEMMS
SGEMM

Figure 3.