The Tradeoff Between Processor Speed and Parallelism for Supercomputers

Min-You Wu

YALEU/DCS/TR-801
June 1990

THE TRADEOFF BETWEEN PROCESSOR SPEED AND PARALLELISM FOR SUPERCOMPUTERS

Min-You Wu

Department of Computer Science

Yale University

New Haven, CT 06520

(203) 432-4761, wu-min-you@cs.yale.edu

Abstract — The spectrum of current supercomputing technologies ranges from a single very fast processor to 64K single-bit processors. What is the best approach for supercomputing? In this paper, we study the relation between performance and cost of single-processor systems, as well as parallel systems. The optimal processor speed and number of processors can be found for either a performance requirement or a given budget. Furthermore, we conclude that the highest performance can be obtained by using a few fastest available processors and the most efficient computation may be obtained by using many fast microprocessors with good parallelization techniques.

Keywords: supercomputers, cost, performance, processor speed, parallel systems, parallelization loss.

1. INTRODUCTION

Mainline supercomputers employ the fastest but very expensive processors with small-scale parallelism. Recently, many people have claimed that they can use hundreds, thousands, even tens of thousands of low-cost processors to achieve the same performance [1, 2, 3]. For GFLOP computation, every approach is used ranging from 64K single-bit processors in the Connection Machine to eight very powerful processors in Cray Y-MP [4]. Although parallelization is already an important technique, it is still not clear how large parallelism can best be used for supercomputing.

In one extreme, some people believe parallelization is a very complex task. Parallel performance will suffer from the non-parallelizable fraction of computation and parallelization overhead. They believe that high performance can be obtained mainly by using the fastest processors with simple vectorization. However, when technology approaches its limit, performance cannot be gained merely by increasing processor speed. Actually, the number of processors in a system is steadily increasing even in mainline supercomputers [5]. In the other extreme, some people believe in massively parallel systems. They have given examples where performance can be increased linearly with the number of processors [6]. Thus, they simply estimate their system performance by multiplying the individual processor performance by the number of processors. However, performance can not be predicted in this way because of parallelization loss. Usually, the aggregate performance drops when the number of processors increases. This performance loss depends on parallelization techniques and the nature of applications.

While some supercomputer users are looking for the highest possible performance, other users prefer effective computation, that is, maximized performance for a given cost or the minimized cost for a performance requirement. Our study concerns both absolute performance and effective computation. These goals may be reached by either more powerful processors or more processors, or both. As single processor technology approaches its physical limit, parallel processing becomes more important. The trade-off between powerfulness and parallelism depends on many factors, such as performance requirements, budget limits, and level of technology.

In this paper, the cost-performance ratio of single processor systems is studied first. The impact of parallelization is illustrated by Amdahl's revised law. Then, we show how to obtain highest performance, and how to maximize performance and reduce cost.

2. COST-PERFORMANCE RATIO

For proper configuration of supercomputers, not only performance but also cost must be considered In each processor technology, the cost of a processor may increase with its performance sublinearly, linearly, or superlinearly. Usually, as shown in Figure 1, the cost increases sublinearly with performance at the low end of the technology. At the high end of the technology, the cost increases superlinearly before reaching the next technology. At the low end of the next technology, the cost increases sublinearly again. For each technology, the lowest cost-performance ratio is at the point where the cost-performance curve changes from sublinear to superlinear.

We are interested in two technologies for supercomputing: microprocessor technology and mainframe technology. The cost for single processor systems is shown in Figure 2.

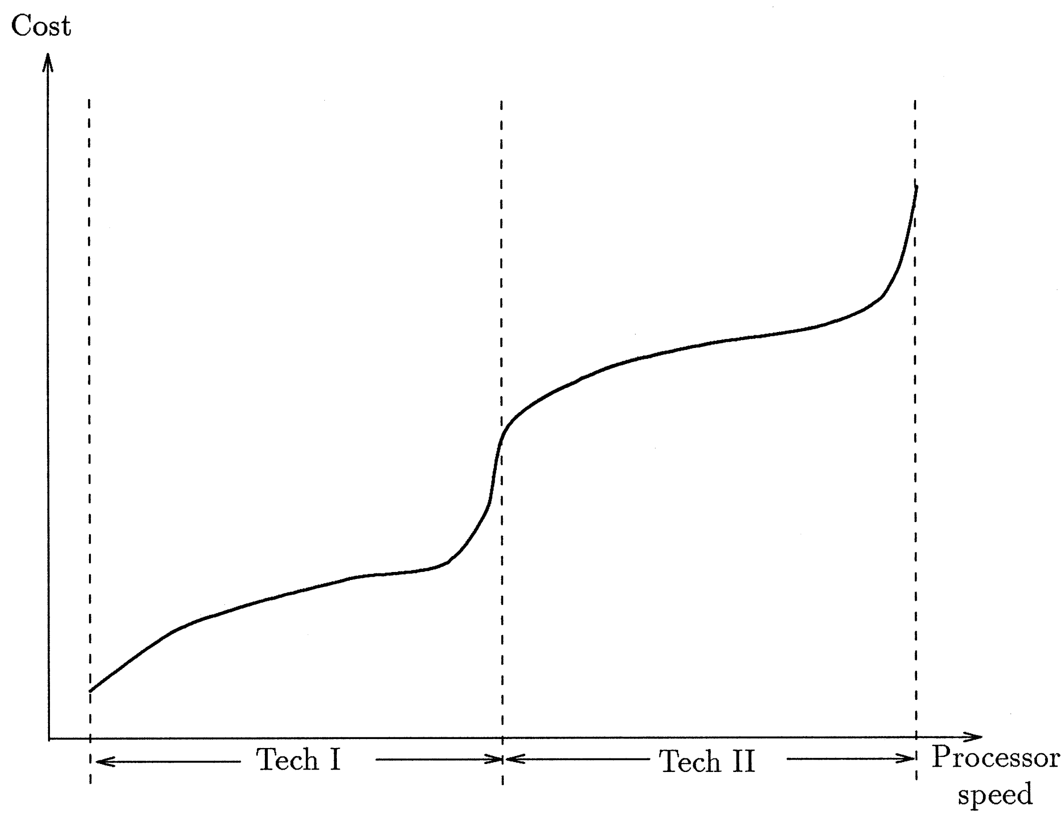

Figure 1: The cost-performance curve of two technologies.

Notice that the cost of the microprocessor here is the system price instead of the processor price. The cost of the microprocessor increases with processor performance sublinearly before 20 MFLOPS. Then the curve increases superlinearly until it meets the next technology. In mainframe technology also, the cost-performance curve is sublinear before reaching the high end of the technology. We have to point out that this curve is only valid at the current level of technology. It will change with the progress of technologies.

The cost-performance ratio is shown in Figure 3. It can be seen that the best cost-performance ratio is about \$1K per MFLOP at 20 MFLOPS. The cost-performance ratio is about \$5K to \$10K per MFLOP for mainframe technology. These ratios will decrease with the progress of technologies. Since mainframe technology is approaching the physical limit, it seems unlikely that the ratio will be lower than that for the microprocessor technology.

Modern supercomputers seldom consist of only one processor. Usually, many processors cooperate to obtain high performance. However, the performance of such a system cannot be calculated by multiplying individual processor performance by the number of processors. The impact of parallelization must be considered.

3. IMPACT OF PARALLELIZATION

The speedup of a parallel system is limited by the sequential part of a program [7], as well as parallelization overhead [8]. Because of these limits, high performance cannot be obtained by simply increasing the number of processors. The impact of parallelization can be modeled by two parameters, the Amdahl fraction a and the parallelization loss coefficient c.

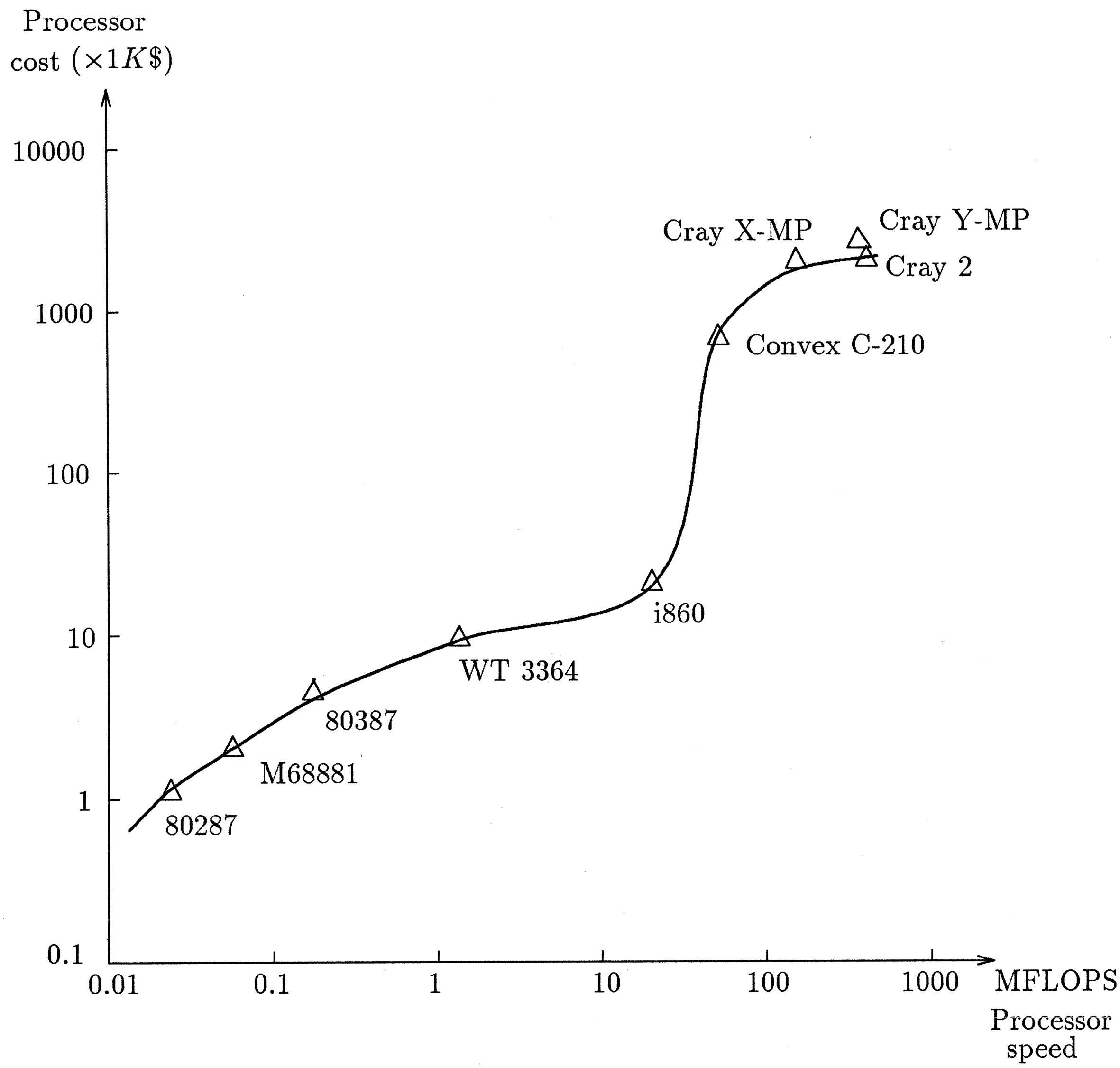

Figure 2: The cost vs. performance of single processor systems.

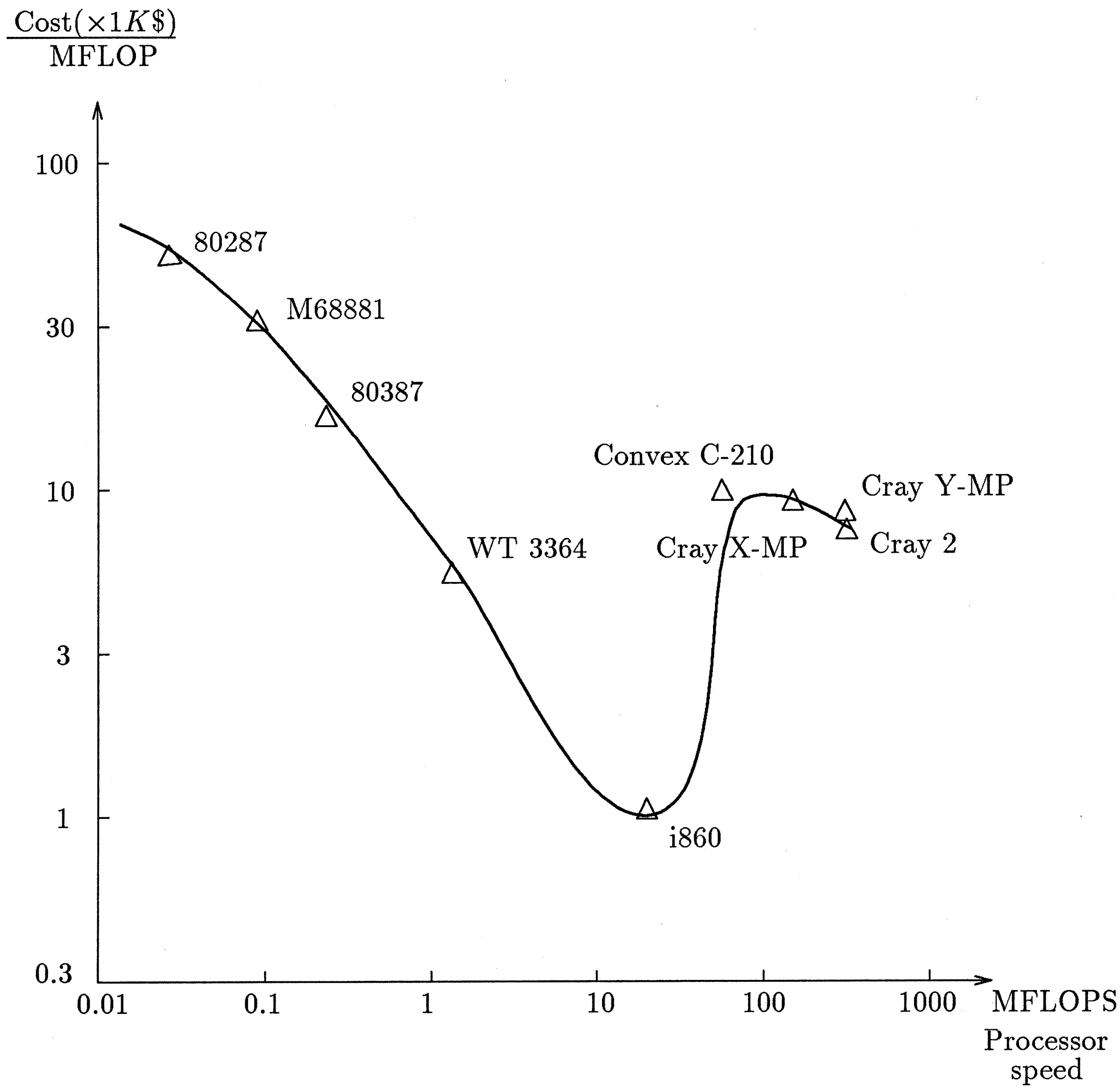

Figure 3: The cost-performance ratios.

In a parallel system, speedup is defined as:

$$S = \frac{T(1)}{T(N)}$$

where $T(1)$ is the execution time on a single processor without overhead, and $T(N)$ is the execution time on N processors. Efficiency is defined as:

$$\mu = \frac{S}{N}$$

where N is the number of processors.

$T(1)$ can be divided into a sequential part T_s and a parallel part T_p :

$$T(1) = T_s + T_p$$

Parallel execution time can be represented as:

$$T(N) = T_s + \frac{T_p}{N} + T_l(N)$$

where $T_l(N)$ is the *parallelization loss* in executing the parallel part of a program.

The Amdahl fraction is:

$$a = \frac{T_s}{T(1)}$$

Similarly, we can define a *parallelization loss coefficient* as:

$$b(N) = \frac{T_l(N)}{T(1)}$$

With the Amdahl fraction and the parallelization loss coefficient, the speedup can be represented as:

$$S = \frac{T(1)}{T_s + \frac{T_p}{N} + T_l(N)} = \frac{N}{1 + a\,(N-1) + b(N)\,N}$$

$b(N)$, the parallelization loss coefficeint, is a complicated function of many factors, which increases with $O(logN)$ in hypercube systems [9]. It can be simply modeled as:

$$b(N) = c \cdot logN$$

where c is the *base parallelization loss coefficeint*, which is equal to $b(2)$, the parallelization loss with two processors.

Then the speedup becomes:

$$S = \frac{N}{1 + a(N-1) + c\,N\,logN}$$

and the efficiency is:

$$\mu = \frac{1}{1 + a(N-1) + c\,N\,logN}$$

The value of c depends on the parallelization loss and problem size. The parallelization loss T_l reflects the combination of system hardware and software quality. The value of T_l is affected by

- communication overhead, latency, and bandwidth;
- bookkeeping overhead and other software overhead; and
- dependencies between different processes.

It includes not only overhead but also the idle time caused by load imbalance of the parallel part of the computation. The value of T_l depends on the parallelization techniques used. The relevant issues include partitioning, scheduling, and synchronization, as well as hardware technology. When parallelization techniques become mature, the value of T_l will

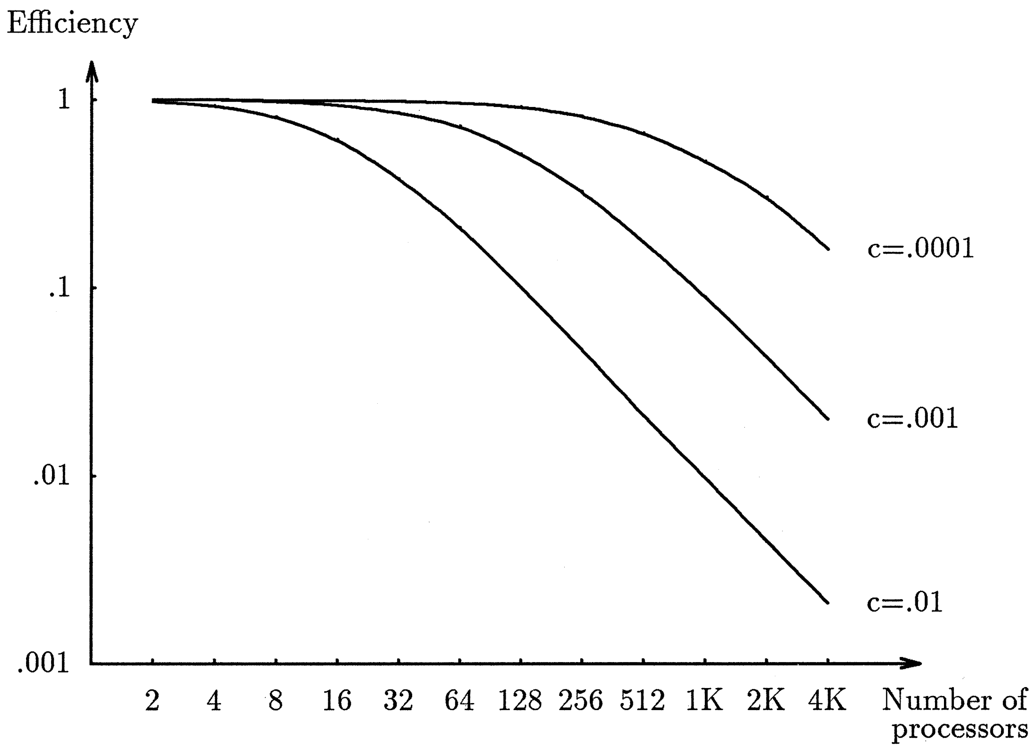

Figure 4: Efficiencies for different values of c.

decrease. At a certain level of parallelization technique, c also depends on problem size. The larger the problem size, the better performance will be.

In the following, we set the value of a, the Amdahl fraction, to 0.0001 and vary c to show different effects of parallelization techniques. In Figure 4, the efficiencies for different values of c are shown with various numbers of processors. The impact of efficiency will be studied in the next section.

4. OPTIMAL PROCESSOR SPEED AND NUMBER OF PROCESSORS

Although the cost-performance ratio reaches its minimum at 20 MFLOPS as shown in

Figure 3, many applications require much higher performance. Some performance requirements, say 10 GFLOPS, cannot be satisfied by using a single powerful processor at the current level of technology. Therefore a parallel system is necessary. However, as discussed previously, a parallel system has an efficiency loss, and the efficiency loss increases with the number of processors. For this reason, we cannot simply expect that 500 20-MFLOP processors or 10,000 1-MFLOP processors will achieve 10 GFLOP performance. We must consider the efficiency, and the performance of a parallel system becomes:

$$V(N) = \mu N V(1)$$

where $V(1)$ and $V(N)$ are the performance of a single processor and N processors, respectively. Besides the parallelism of applications, efficiency μ depends on the parallelization loss coefficient, c. The smaller the value of c is, the better the parallelization technique is. The effect of c on the cost-performance ratio is shown in Figure 5. The curves represents the costs for 300 MFLOPS performance, on the assumption that the cost is proportional to the number of processors. All the curves except the lowest one assume the value of the Amdahl fraction a to be 0.0001. The lowest curve is the ideal case with $a = 0$ and $c = 0$. The cost increases with the value of c. When c becomes large, the advantages of low-price microprocessors will disappear due to large parallelization loss. In other words, a parallel system built of many small processors can deliver high performance with good cost-performance ratio only when the parallelization technique is satisfactory. Furthermore, there is an additional cost for rewriting the existing code for parallel systems.

In many circumstances, we are given a fixed budget instead of a performance requirement. The goal is to maximize performance with the budget. In Figure 6, the budget is fixed to 2.5 million dollars. It shows that one GFLOP can be obtained with many small processors if we apply a good parallelization technique. However, if the parallelization

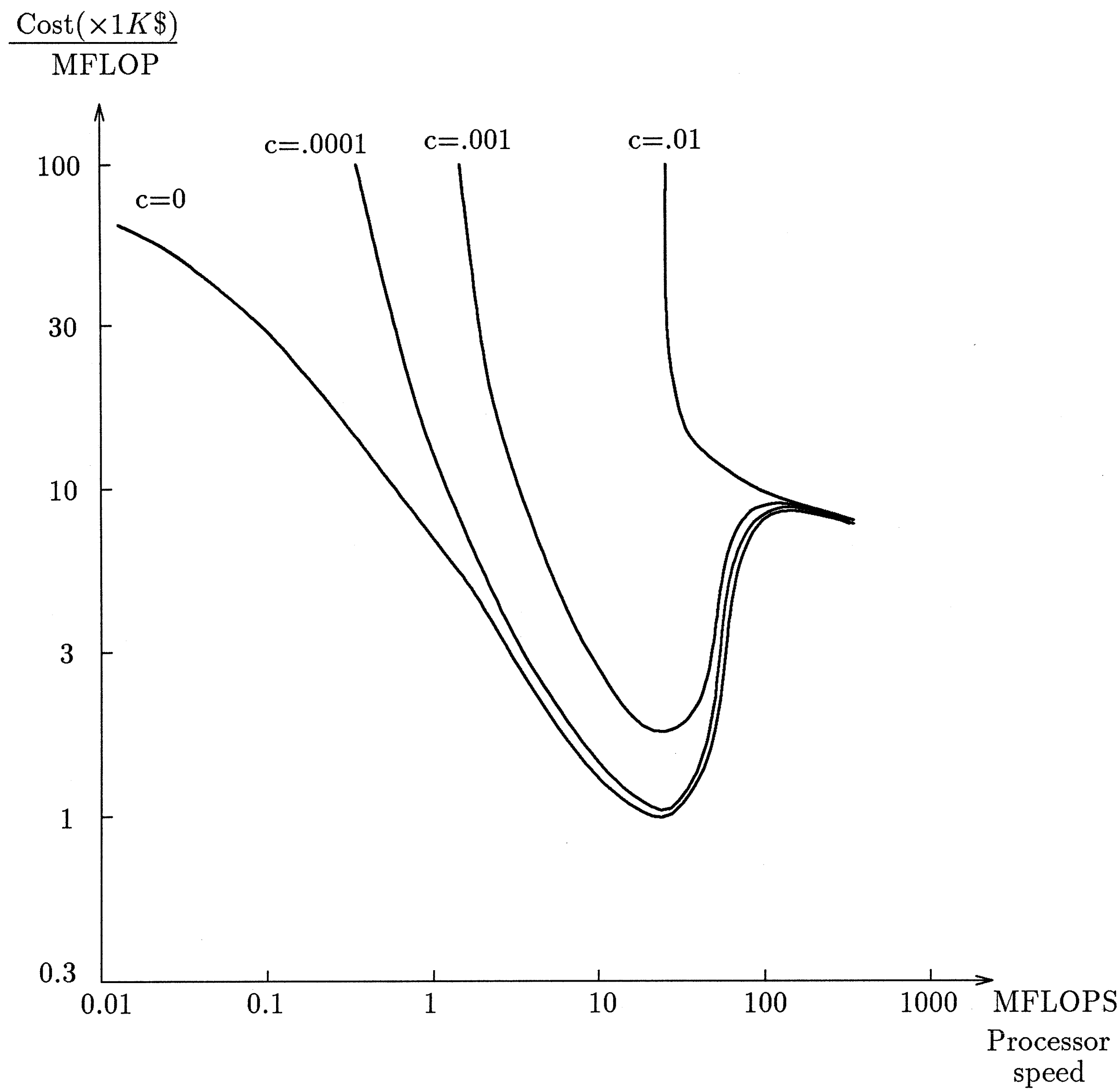

Figure 5: The costs for a performance requirement with different values of c.

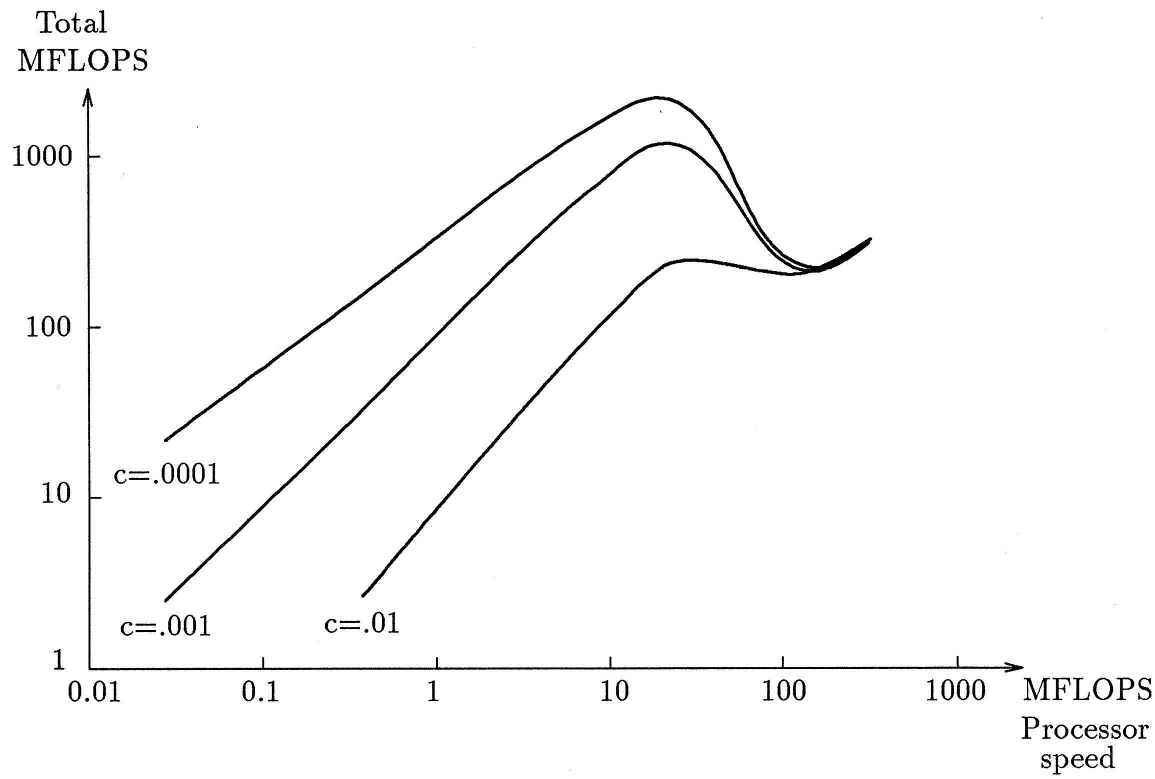

Figure 6: The performance for a fixed cost.

technique is not satisfactory, say $c = 0.01$, the best performance of 300 MFLOPS will be obtained by a single processor instead of a parallel system. It confirms that performance of a parallel system depends heavily on the parallelization technique. Before parallelization techniques become mature, using less processors may be the best choice for many users.

An interesting fact is that if we have an unlimited budget, the highest performance is obtained by using a small scale parallel system consisting of the fastest processors. Figure 7 shows when the budget increases, the advantage of using parallel systems declines. Thus, investment in increased numbers of processors cannot always be exchanged for higher performance. With the exception of certain applications, system performance is limited by the speed of processors in general. When the highest possible performance is required, fast processors are necessary. Some massively parallel machines consist of large numbers of single-bit processors. The single-bit approach suffers in low-parallelism parts of programs, for example, an operation executed on a single column of a matrix. When there is not enough parallelism, this approach may cause serious load imbalance. As an example, when executing an operation on a column of a 2K $\times$ 2K matrix on 64K single-bit processors, 31/32 processors are idle. However, if it is executed on 2K 32-bit processors, all processors are busy and the execution time is reduced by a factor of 32. The single-bit approach does not utilize the available parallelism in application problems. Therefore, this approach is not appropriate for general-purpose supercomputing, except for some special purpose applications, such as image processing.

Figures 8 and 9 show the relation of performance, cost, and the number of processors for $c = 0.001$ and 0.0001, respectively. From the figures, one can see the relation between performance and cost for a given number of processors. Furthermore, for a given cost, the best configuration can be found for the highest achievable performance. For example, with a budget of one million dollars, 64 20-MFLOP processors can deliver 800 MFLOPS when c=0.001, or 1.2 GFLOPS when c=0.0001. With thirty million dollars, 1K 20-MFLOP processors gives 10 GFLOPS when c=0.0001. However, if parallelization techniques are not good enough, say c=0.001, one may select 16 300-MFLOP processors for 4.5 GFLOPS. The figures can also be used to find the lowest price for a required performance.

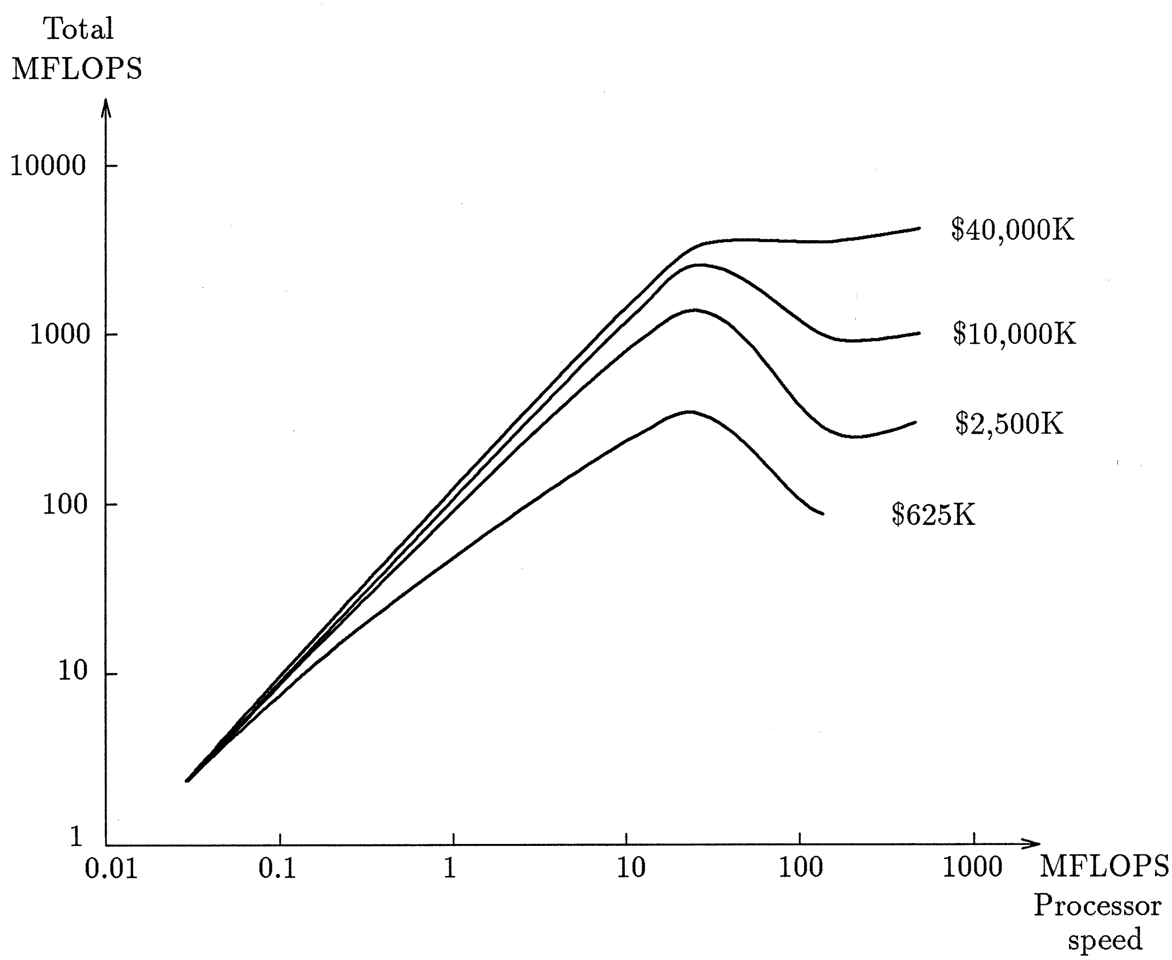

Figure 7: The achievable performance for different costs.

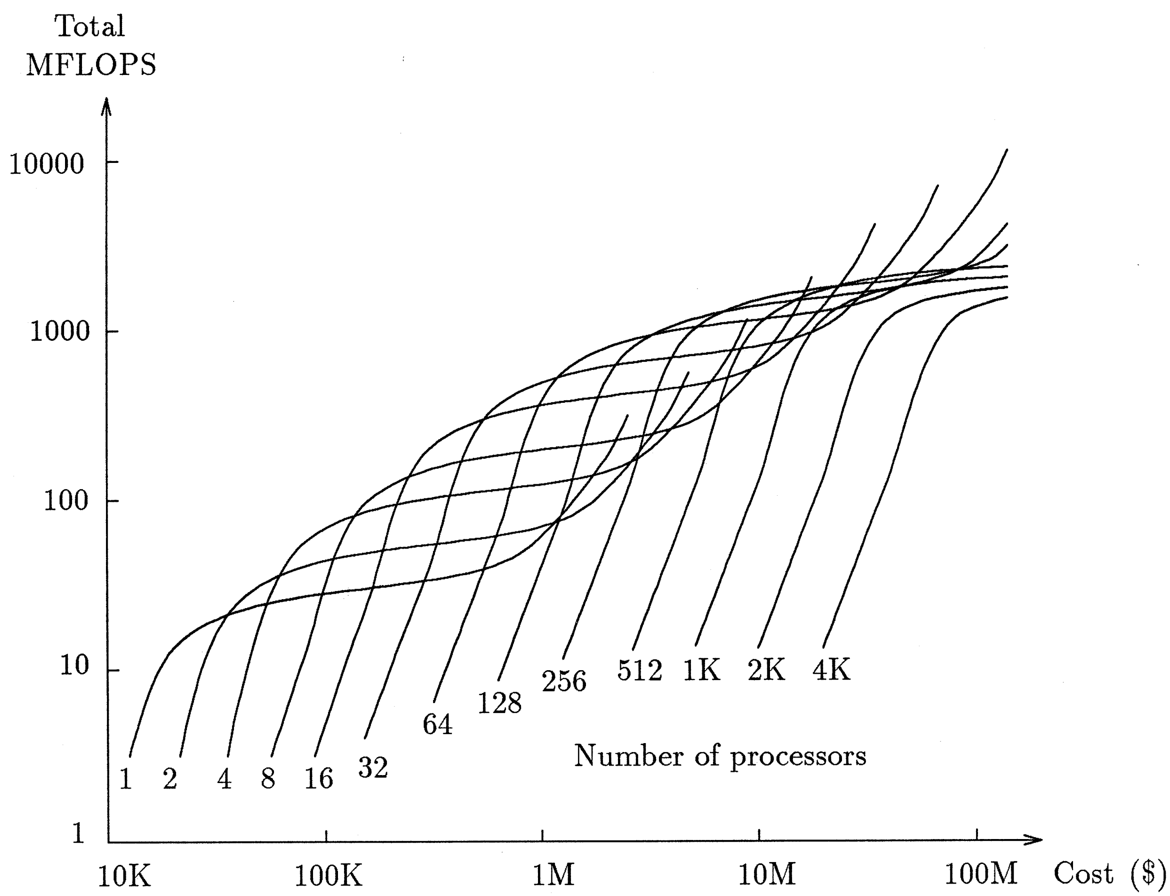

Figure 8: The relation of performance, cost, and number of processors (c=.001).

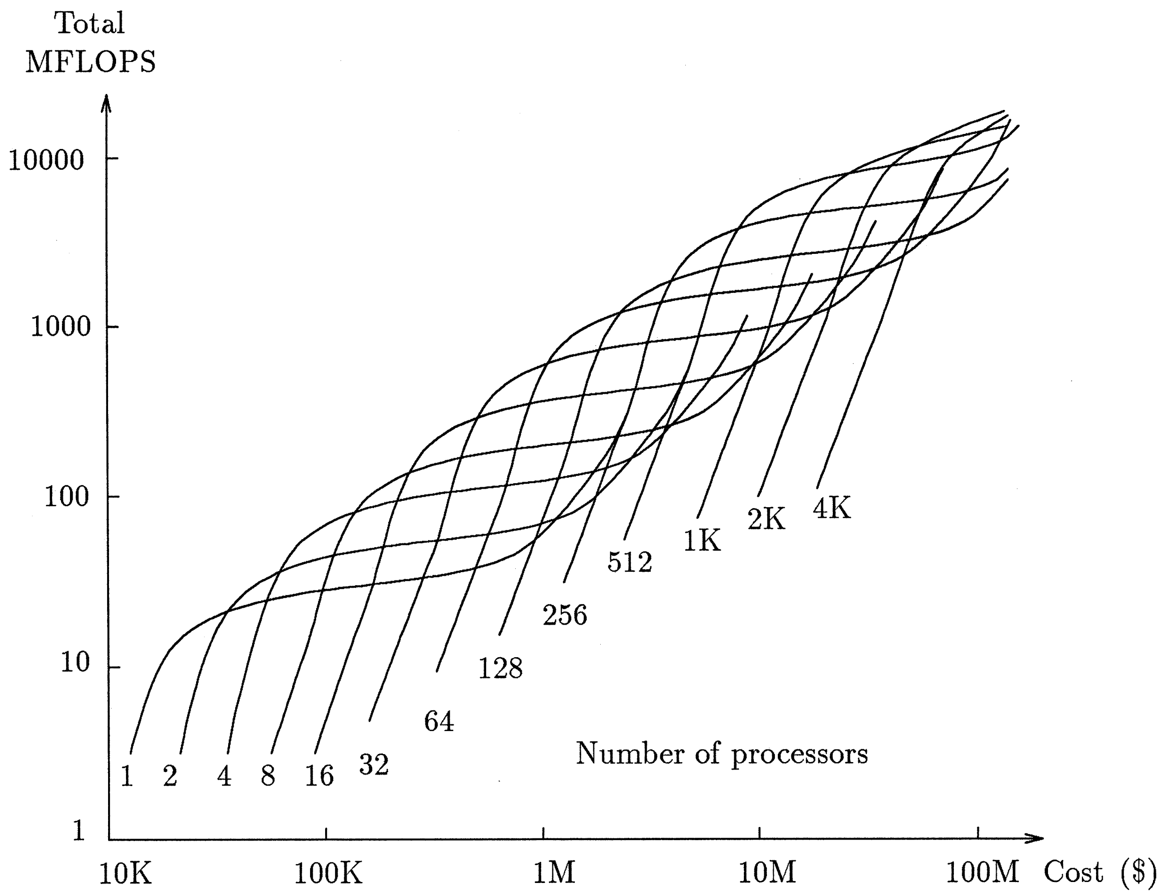

Figure 9: The relation of performance, cost, and number of processors (c=.0001).

5. CONCLUSION

Parallelization techniques play an important role in supercomputing. A parallel system can potentially deliver high performance at low cost. However, the success of parallel systems depends heavily on the progress of parallelization techniques. The parallelization loss coefficient c has been defined to represent the impact of parallelization techniques. The value of c decreases when the parallelization techniques, including hardware, software, and application techniques, become mature. The value of c can also be reduced by running problems of large size. Since the problem size running on supercomputers is usually large, we can expect small c. With a small value of c, a parallel system consisting of many microprocessors can deliver higher performance than mainframe systems. For best performance, there is an optimal tradeoff between processor speed and number of processors. Furthermore, system performance will always be limited to some extent by the speed of individual processors. In general, we should not expect a huge number of slow processors to deliver very high performance. Fast processors are necessary for high speed computing even in a parallel system.

Acknowledgments

The author wishes to thank Joe Rodrigue for his editorial efforts. The generous support of the office of Naval Research under Contract N00014-86-K-0310 is gratefully acknowledged.

References

[1] J. P. Hayes, T. Mudge, Q. F. Stout, S. Colley, and et al. A microprocessor-based hypercube supercomputer. *IEEE Micro*, 6:6–17, October 1986.

[2] L. W. Tucker and G. G. Robertson. "Architecture and applications of the Connection Machine". *IEEE Computer*, 21(8):26–38, August 1988.

[3] J. L. Potter and W. C. Meilander. Array processor supercomputers. *IEEE Proceedings*, 77(12):1896–1914, December 1989.

[4] T. S. Perry and G. Zorpette. Supercomputers experts predict expansive growth. *IEEE Spectrum*, 26(2):26–33, February 1989.

[5] Gordon Bell. The future of high performance computers in science and engineering. *Communications of the ACM*, 32(9):1091–1101, September 1989.

[6] J. L. Gustafson and G. R. Montry. Programming and performance on a cube-connected architecture. In *IEEE COMPCON*, pages 97–100, March 1988.

[7] G. M. Amdahl. Validity of the single processor approach to achieving large scale computing capability. In *Proc. AFIPS Comput. Conf.*, volume 30, pages 97–100, 1967.

[8] C. Lazou. *Supercomputers and their Use.* Oxford Science Publications, 1988.

[9] M. L. Barton and G. R. Withers. Computing performance as a function of the speed, quantity, and cost of the processors. In *Supercomputing '89*, pages 759–764, November 1989.