LA-UR -83-1389

MASTER

CONF-830963--1

Los Alamos National Laboratory is operated by the University of California for the United States Department of Energy under contract W-7405-ENG-36

TITLE DRAFT REMARKS FOR THE IFIPS CONGRESS '83 PANEL ON HOW TO OBTAIN HIGH PERFORMANCE FOR HIGH-SPEED PROCESSORS

AUTHOR(S) B. L. Buzbee

LA-UR--83-1389

DE83 012669

SUBMITTED TO IFIPS Congress '83
Paris, France
September 19-23, 1983

DISCLAIMER

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

The Los Alamos National Laboratory requests that the publisher identify this article as work performed under the auspices of the U.S. Department of Energy

Los Alamos Los Alamos National Laboratory
Los Alamos, New Mexico 87545

DISTRIBUTION OF THIS DOCUMENT IS UNLIMITED

DRAFT REMARKS FOR THE IFIPS CONGRESS '83 PANEL ON HOW TO OBTAIN HIGH PERFORMANCE FOR HIGH SPEED PROCESSORS

By

B. L. Buzbee*

Computing Division
Los Alamos National Laboratory
Los Alamos, NM, USA

DISCLAIMER

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

*This work done under the auspices of the U.S. Department of Energy.

High speed processors play important roles in all areas of science but are most often used in the execution of large-scale numerical simulations. In this role they help scientists and engineers gain insight by

- enabling them to treat complexity in models that is not otherwise tractable,
- enabling them to study phenomena that are not feasible to study in the laboratory, and
- helping to test theory.

This ability to help the scientist gain insight constitutes the greatest value of computers and provides the primary motivation for this panel.

Historically, scientists engaged in modeling have constrained their numerical simulations so that the average execution time is about ten hours. This constraint reflects the scientist's need to make daily progress. Thus, the amount of complexity incorporated in models is limited by the associated computer's ability to produce results in about a ten-hour execution time. This limitation combined with a computer's ability to help the scientist gain insight causes us to continually seek bigger and faster computers.

TRENDS

The growth rate in execution bandwidth of high speed processors is diminishing. This is illustrated in Figure 1, which shows the execution bandwidth of some high speed processors over the era of electronic computation. These data have been approximated by a modified Gompertz curve, and the asymptote to that curve is about 3 billion operations per second. Note that the Cray-1 is already within an order of magnitude of that asymptote. Of course, we must ask if the curve will accurately forecast performance in view of new and exciting developments in very large-scale integrated circuit technology. Dr. Takamitsu Tsuchimoto, Fujitsu Limited, will discuss this and related issues. If, indeed, this Gompertz curve continues to forecast the future accurately, then we are left with the unpleasant conclusion that a single processor has a maximum performance level and that state-of-the-art equipment is approaching it. An obvious way to circumvent this maximum is through the use of parallelism. This panel will also address some potentials and problems of parallel processing.

Professor Arvind, Massachusetts Institute of Technology, will discuss some architectural and performance issues that arise in parallel processing architectures that are based on von Neumann style uniprocessors and will sketch datablow solutions to them. Professor S. Levialdi, Instituto Scienze dell'Informazione, Bari, Italy, will review some parallel computer architectures for image processing. Dr. V. Kotov, Novosibirsk, USSR, will discuss interrelations between computer technology, architecture, programming languages, and algorithms. In the next few minutes I will take a brief look at systems of a few tightly coupled processors. In the course of my remarks I hope to show that algorithms and systems-related issues will be critical to the overall success of tightly coupled systems.

ASYNCHRONOUS SYSTEMS OF A FEW TIGHTLY COUPLED HIGH SPEED PROCESSORS

Asynchronous systems of a few tightly coupled high speed processors are a natural evolution from high speed uniprocessor systems. Indeed, a system with 2-4 processors will soon be available, e.g., the Cray XMP and the Cray-2. Systems with 8-16 processors are likely by the early 1990's. What are the prospects of using the parallelism in such systems to achieve high speed in the execution of a single application? Answering this question is an exercise in research. The remainder of this paper will discuss some of the associated issues.

The key issue in parallel processing a single application is speedup as a function of the number of processors used. We define speedup as

$$S_p = \frac{\textit{execution time using one processor}}{\textit{execution time using p processors}}$$

THE WARE MODEL

To estimate performance of a tightly coupled system on a single application, we use a model of parallel computation introduced by Ware [1].

Let

$p = \textit{number of processors},$

and

α = *percent of parallel processable work in the application.*

Assume at any instant that either all p processors are operating or only one processor is operating. If we normalize the execution time using one processor to unity, then

$$S_p = \frac{1}{(1-\alpha) + \frac{\alpha}{p}} \ .$$

Also

$$\frac{dS_p}{d\alpha}\Big|_{\alpha=1} = p^2 - p \ .$$

Figure two shows the Ware model of speedup as a function of α for a 4-processor, an 8-processor, and a 16-processor system. The quadratic behavior of the derivative is dramatic and results in low speedup for α less than .9. Consequently to achieve significant speedup, we must have highly parallel algorithms. It is by no means evident that algorithms in current use on uniprocessors contain the requisite parallelism. In cases where they do not, research will be required to find suitable replacements. When highly parallel algorithms are available they must be implemented (combined) with care because α spans the entire application. Quadratic behavior of the derivative at large α means that a small change in it produces a large change in speedup.

Those who have experience with vector processors will note a striking similarity between the Ware curves and models of vector performance where the abscissa is the percent of total vectorizable computation. This is because the assumption of the Ware model implies a two-state machine, that is, in one state only one processor works and in the other state all p processors work. A vector processor can also be viewed as a two-state machine. In one state it is a relatively slow general purpose machine, and in the other state it is capable of high performance on vector operations. Thus figure 2 also gives the performance of vector processors where p is the relative performance of the vector and scalar states.

THE MODIFIED WARE MODEL

Ware's model is inadequate in that it assumes that exactly the same instruction stream will be executed on a parallel system that is executed on a single processor, and, thus, that the same amount is of work is done in both. Seldom is this the case because synchronization and communication in asynchronous systems usually require execution of instructions that are not present in a uniprocessor implementation. Further, parallel algorithms may require additional instructions. To correct for this, we add a term, $\sigma(p)$, to the execution time for parallel implementation. σ is at best nonnegative,and usually monotonically increasing with p. Actually, it is a function not only of p, but of the algorithm and the architecture, even of α. Let $\tilde{S}_p$ denote the modified model, then

$$\tilde{S}_p = \frac{1}{(1-\alpha) + \frac{\alpha}{p} + \sigma(p)}$$

$$= \frac{p}{1+p\ \sigma(p)} \ at\ \alpha = 1 \ .$$

Consequently in general the maximum speedup of a real system will be less than p, and it may be significantly less (note also that $\tilde{S}_p < 1$ for small α). Further, $\tilde{S}_p$ will have a maximum for sufficiently large p, i.e., $\frac{\alpha}{p}$ becomes insignificant while $\sigma(p)$ continues to increase. Thus another research opportunity involves finding algorithms, programming languages, and parallel architectures that, when used as a system, yield "small" σ's.

An important secondary question is how does one determine that a good $\sigma(p)$ has been achieved? Note that

$$\frac{1}{\tilde{S}_p} = \frac{1}{p}\left[p(1-d) + \alpha + p\sigma(p) \right]$$

$$\cong \frac{1}{p} \text{ if } \alpha \ large \text{ and } \sigma(p)\ small$$

$\tilde{S}_p$ is proportional to execution time in parallel mode. So a plot of execution time versus $\frac{1}{p}$ will reveal when high α and low σ have been achieved. For example researchers at Los Alamos are engaged in efforts to parallel process several generic classes of scientific computation. One of these classes is particle-in-cell simulations that are widely used in plasma modeling. Figure 3 shows the results. Similar results have been obtained for fluid flow models and Monte Carlo simulations for up to $p = 8$.

NONREPEATABILITY

Throughout the era of Von Neumann architecture we have enjoyed repeatability in computation, i.e., repeated computation with invariant code and data yields invariant results. Repeatability is not assured on an asynchronous system because the precise sequence of computation depends on the temporal correlation of processor activities. Absence of repeatability has already been manifest in Monte Carlo simulations [2] and may have important consequences in other algorithms, e.g., roundoff error critical algorithms. Debugging is definitely complicated by nonrepeatability. Similar difficulties are well known to developers of computer network software, and programming languages that permit side affects will provide one way of creating them. The result may be an increased interest in new programming languages, code management tools, etc., for specific use in parallel processing.

FUTURE RESEARCH

We began this discussion by asking about the possibility of gaining higher speed by exploiting parallelism in asynchronous systems of a few tightly coupled processors. The question at present is not answerable. In 1970 Minsky [3] conjectured that average speedup in parallel systems would be proportional to *logp*. Because of economic and performance considerations this result would be unacceptable in the systems under discussion. Yet Ware's model confirms that to do better than Minsky's conjecture will require highly parallel algorithms. Further, they must be insensitive to nonrepeatability. The modified Ware model indicates that they must be supported by efficient systems and programming languages. Other studies [4,5] suggest that average speedup might be proportional to *p/logp*. Regrettably there is a paucity of experimental data with which to validate or invalidate these and similar studies. This is a great misfortune. A recent study noted [6]

> "There is an abundance of concepts for parallel computing and abundant opportunities for developing these concepts and their applications. The current bottleneck to progress is the difficulty of executing significant experimental studies. These studies are essential to evaluation of total system concepts."

Thus there is need for experimental asynchronous systems and data on the use thereof.

SUMMARY

Systems of a few tightly coupled high performance processors have the potential to provide significant increases in computational capability. Realizing this potential will require development of highly parallel algorithms. These must be combined with suitable programming languages and architectures such that the overall implementation introduces little additional work relative to uniprocessor implementation. Experimentally validated models of performance will facilitate this research. In general, availability of experimental equipment will be a pacing factor in research on asynchronous systems.

REFERENCES

(1) W. Ware, "The Ultimate Computer," *IEEE Spect,* March, 1973, pp. 89-91.

(2) P. Frederickson, R. Hiromoto, T. Jordan, B. Smith, and A. Warnuk, "Pseudo-Random Trees in Monte Carlo," Los Alamos Technical Note LA-UR-83-1130, April, 1983.

(3) M. Minsky, "Form and Content in Computer Science," ACM Lecture, *JACM* 17, 1970, pp. 197-215.

(4) R. Lee, "Performance Bounds in Parallel Processor Organizations," in *High-Speed Computer and Algorithm Organization,* Kuck, Lawrie, and Sameh, Eds., Academic Press: New York, 1977, pp. 453-455.

(5) D. Kuck, et al, "Measurements of Parallelism in Ordinary Fortran Programs", 1973 Sagamore Computer Conference on Parallel Processing.

(6) "Highly Parallel Computing," A Report to the Information Technology Workshop, J. C. Browne, Chairman, in preparation, Stevens Institute of Technology.

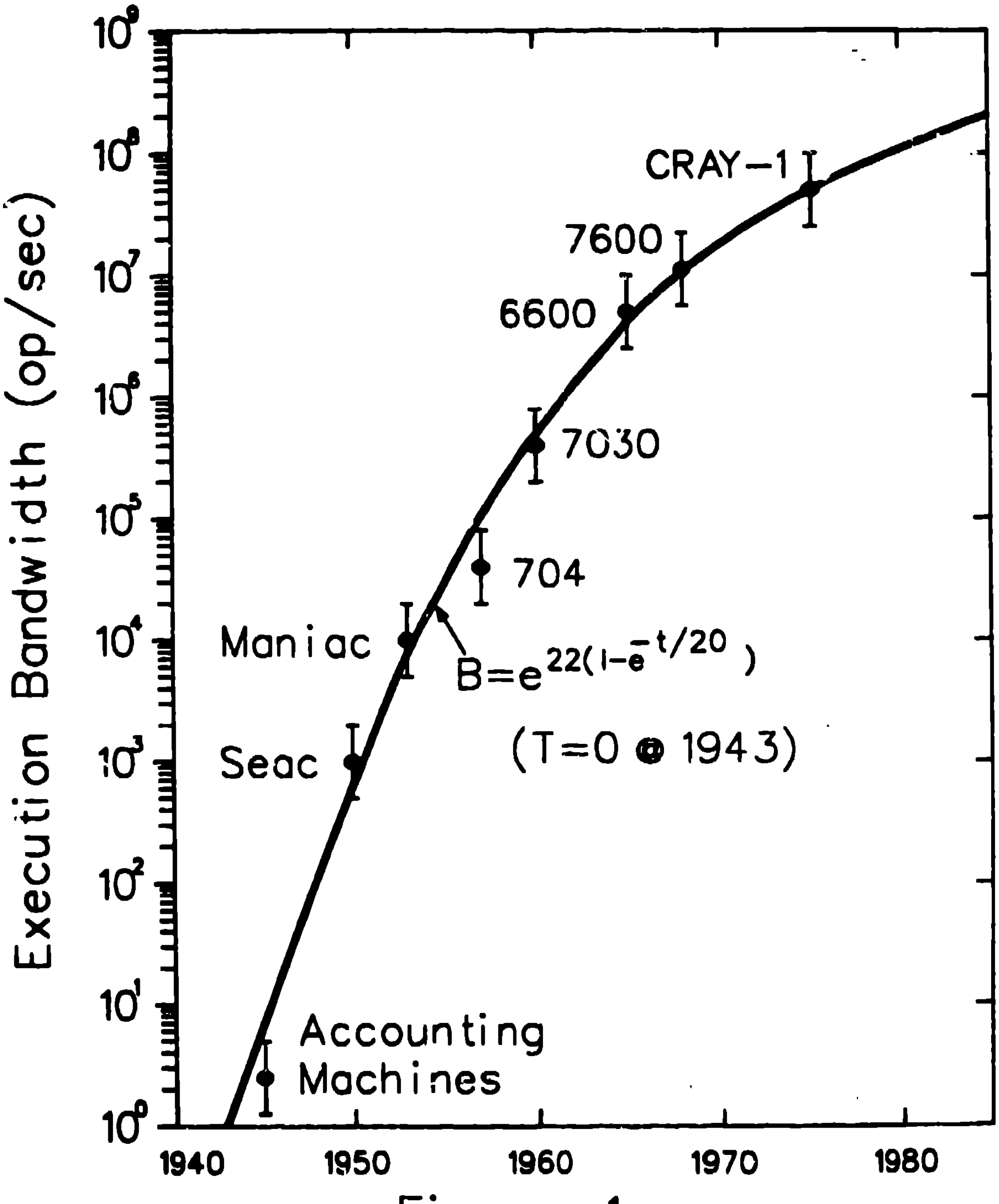

Figure 1.
Trend in execution bandwidth

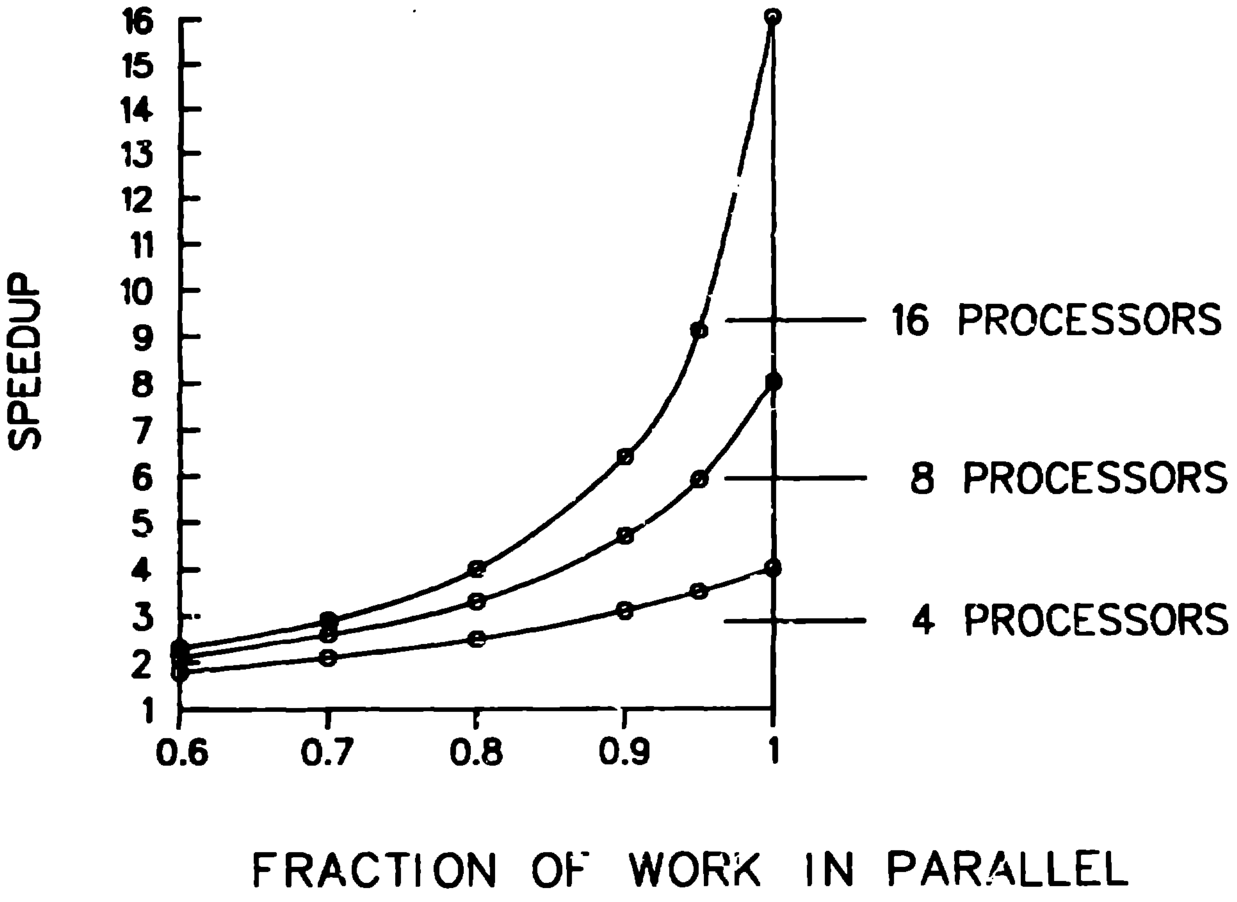

Figure 2. Speedup as a function of parallelism and number of processors.

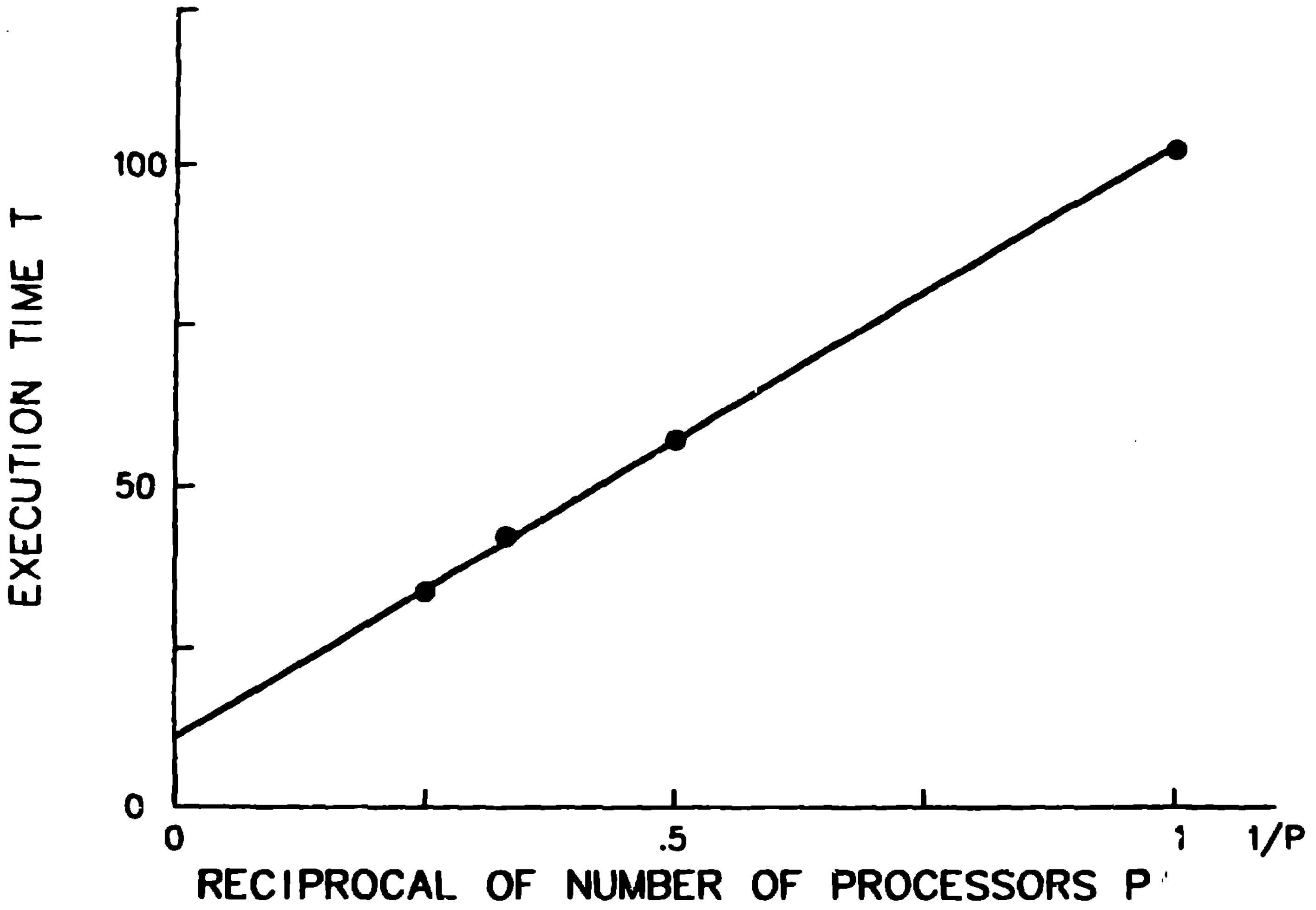

Figure 3. Execution time T of PIC code on one to four processors of UNIVAC 1100/84 as a function of the number of working processors P.