EMPIRICAL RESULTS ON THE SPEED, EFFICIENCY, REDUNDANCY AND QUALITY OF PARALLEL COMPUTATIONS*

Ruby Bei-Loh Lee
Computer Systems Laboratory
Stanford University
Stanford, California 94305.

Abstract — The purpose of this paper is to present empirical results on the performance of parallel computations, with respect to various performance criteria, under different assumptions of the underlying computer architecture. The performance criteria used are the Parallel Index, the Speedup, the Utilization, the Efficiency, the Redundancy, the Compression and a definition of the Quality of the resultant computation. The underlying architectures assumed are parallel processor organizations of both the SIMD and MIMD varieties, with limited and unlimited degrees of physical parallelism.

1 Introduction

Computer architectures incorporating multiple processors which execute in parallel are being designed to speed up the execution-time, for better cost-performance, greater reliability and modularity. The trends appear to be towards special purpose, scientific supercomputers on the one hand, and towards general purpose multiple microprocessor systems with high performance to cost ratios, on the other hand. Although the decreasing cost and size of processors makes it feasible to consider using a large number of processors in a computer organization even at reduced efficiency of each component processor, it is important to estimate the effective speed actually attainable, over a representative set of computations. The sample considered in this paper may be described as existing general technical computations drawn from military, commercial and academic environments. It is emphasized that we are not interested in the maximum or minimum performance for any individual computation, but in the average performance over all the computations.

A computer organization with p parallel processors will rarely attain its maximum parallel execution bandwidth of p operations per time-unit, or a speedup in execution-time of p times that of the uniprocessor organization, due to both logical and physical constraints on the parallel execution of operations. Logical constraints on parallelism include intrinsic data-dependencies, control dependencies and operator precedences in the program, which force a sequential chain of execution amongst the dependent operations, and hence limit the number of operations which may be executed in parallel [9]. Physical constraints on parallelism include the maximum number of processors available in the architecture, the control restrictions on the different types of operations which may be executed simultaneously, and the delays due to the communication and competition amongst the interacting components in the computer organization. The empirical results in this paper account for the logical constraints and the first two physical constraints mentioned. For tractability, the experiments assume that the rest of the system, like effective memory-processor and processor-processor bandwidths, are balanced with respect to the execution-bandwidth of the parallel processors. This may even be considered an advantage since the results are then independent of the specific machine implementation. The empirical performance results in this paper should be interpreted as the best performance results expected with current techniques for parallelism exposure in Fortran programs [5-8, 2], assuming no delays due to the cooperation and competition amongst the components of the parallel processor organization. The results would be degraded if communication delays within the system are considered, but at the same time, the results would probably be improved by the explicit specification of parallel programs or by even better algorithms for the automatic conversion of serial programs to parallel computations.

Some of the questions we ask are: What is the performance of a computer organization with a limited number of parallel processors in the architecture? What is the performance in the idealized case where the number of processors is essentially unlimited? Are there severe performance degradations when only one type of operation may be executed simultaneously in one time-unit by the active processors?

2 Model and Definitions

A parallel computation is a sequence of steps, where each step consists of i operations which may be executed simultaneously, by i parallel processors. A step with i simultaneous operations is said to have degree of parallelism i, $1 \le i \le P$, where P is the maximum degree of parallelism in any step of the computation. The logical parallelism or minimax degree of parallelism, P', is the smallest maximum number of processors required by the computation in order to achieve its minimum execution time, T_{min}.

*This work was supported in part by the Office of Naval Research under Contract N00014-79-C-0516.

CH1569-3/80/0000-0091$00.75

A parallel processor organization is a computer organization with multiple processors, each of which is capable of executing one operation in one time-unit. Each processor is also capable of executing the whole repetoire of operations. An SIMD (Single Instruction Multiple Data) organization is a parallel processor organization where only one type of operation may be executed by the active processors in any one time-unit. An MIMD (Multiple Instruction Multiple Data) organization is a parallel processor organization where more than one type of operation may be executed by different processors in the same time-unit [3].

A parallel processor organization with p processors available in the architecture is said to have limited physical parallelism of degree p, and denoted a p-limited architecture. A parallel processor organization which always has as many processors, P', as required by the computation in order to achieve its minimum execution-time is said to have unlimited physical parallelism, and denoted an unlimited architecture. A p-limited computation is a parallel computation executing on a p-limited architecture, and an unlimited computation is a computation executing on an unlimited architecture.

TOP-form. The TOP-form is a canonic form of parallel computations defined as the following 3-tuple:

$$(T(P), O(P), P)$$

where T(P) is the execution-time of the computation in steps, O(P) is the computation-size in number of operations executed, and P is the maximum degree of parallelism in the computation. P=P', the logical parallelism, for unlimited computations, and P=min(P',p) for p-limited computations.

The TOP-form captures the fundamental difference in the dimensions of the parallel computation when compared to a serial computation. In a serial computation, since each operation takes one time-unit for execution, the execution-time and computation-size have the same value, T(1)=O(1). But in a parallel computation where P>1, the execution-time and computation-size necessarily have different values, T(P)<O(P), forming two distinct dimensions of a parallel computation. The maximum degree of parallelism forms a third variable dimension in a parallel computation.

Equivalence, Optimality and Acceptability. Computations are said to be equivalent if given the same inputs, they always produce the same outputs. The internal algorithms and intermediate results in equivalent computations need not be the same.

An optimal serial computation is defined as a serial computation with the minimum computation-size, Omin. An optimal parallel computation is defined as a minimum-time minimax-parallel computation, which is a computation that achieves the minimum execution time, Tmin, using the minimax degree of parallelism, P'. Further discussion on these definitions of optimality are available in [9].

A serial computation is said to be acceptable for comparison with a parallel computation if O(1)≤O(P). Otherwise, the O(P) operations of the parallel computation may be executed one at a time to obtain a shorter serial execution time and computation size. A parallel computation is said to be acceptable for comparison with a serial computation if T(P)≤T(1). Otherwise, the O(1)=T(1) operations of the serial computation may be executed using one processor to get a shorter parallel execution time. Hence, we propose the following:

Principle of Acceptable Parallel-Serial Comparisons. A performance comparison of a parallel computation with an equivalent serial computation is said to be acceptable iff

$T(P) \leq T(1)$ and $O(1) \leq O(P)$, or equivalently

$T(P) \leq O(1) \leq O(P)$, if each operation takes one time-unit for execution.

This principle of acceptable parallel-serial comparisons is necessary to ensure that any measured performance improvements are due solely to parallel versus serial processing, rather than due to other factors. For example, the Speedup in execution time may be greater than p, the number of processors available in the architecture, when a parallel computation is compared with an unacceptable (nonoptimal) equivalent serial computation. Part of the performance improvement in this case is due to the optimization of a relatively inefficient serial computation. Similarly, the Speedup in execution time may be less than one, if the parallel computation entering into the comparison is unacceptable.

Performance Measures. The TOP-form of a parallel computation and its equivalent serial size form the smallest set of parameters for the evaluation of all the performance criteria considered in this paper. Basically, the performance criteria fall into four categories: the speed of execution given by the Parallel Index and Speedup measures, the utilization of the processor-time resource given by the Utilization and Efficiency measures, the Compression (or conversely, the Redundancy) in the size of the computation, and the resultant Quality of processing.

The Parallel Index and Speedup measure the average and effective speed, respectively, of the parallel computation in operations executed per time-unit:

$$PI(P) = O(P)/T(P),$$
$$S(P,1) = O(1)/T(P) = T(1)/T(P).$$

The Speedup is defined with respect to the computation-size of an equivalent serial computation, and takes into account the extra operations introduced into a parallel computation to reduce its execution time. The Speedup may also be regarded as the ratio of the execution-time of the serial computation, to that of the parallel computation, making it equivalent to the definition found in [5].

The Parallel Index and Speedup may also be regarded as measures of the average and effective parallel execution bandwidths of the underlying parallel processor organization, during the execution of the given computation.

The Utilization and Efficiency measure the cost-effectiveness of the computation in the sense that they weigh the speed improvement with the number of processors required. They measure the performance of the parallel computation with respect to its use of the processor-time resource:

$$U(P) = O(P)/[P.T(P)], \quad E(P,1) = O(1)/[P.T(P)].$$

In figure 1, the Utilization is the proportion of the rectangle PxT(P) covered by busy processor-steps, i.e., those time-units where a processor is busy executing an operation. The Efficiency may be regarded as the ratio of the serial processor-time requirement over the parallel processor-time requirement, since T(1)=O(1).

The Redundancy measure is the ratio of the parallel computation-size, O(P), to the serial computation-size, O(1), of an equivalent serial computation. The Compression measure is the inverse ratio:

$$R(P,1) = O(P)/O(1), \quad \text{and} \quad C(P,1) = O(1)/O(P).$$

One significance of the Redundancy measure is that it relates the relative speed and efficiency measures, S and E, to the absolute speed and efficiency measures, PI and U:

$$S = C.PI = PI/R \quad \text{and} \quad E = C.U = U/R$$

The Speedup, Efficiency and Compression measures compare serial to parallel execution-time requirements, processor-time requirements and computation-size requirements, respectively (see table 1). In an optimal serial computation, the execution-time, processor-time and computation-size requirement are each equal to Omin. The Quality measure is defined as an overall performance measure comparing serial to parallel computations with respect to these three requirements:

$$Q(P,1) = S.E.C = S.E/R = \frac{Omin^3}{T(P)^2.O(P).P}$$

Hence, the Quality measure is a more stringent measure of the performance improvement of parallel versus serial processing than the Speedup measure. One use of the Quality measure is to decide whether parallel processing is preferable to serial processing for a given program. For example, a computer installation may decide that parallel processing is desirable for a given program if the quality of processing increases by at least fifty percent ($Q \geq 1.5$).

In all acceptable parallel-serial comparisons, the following relationships hold:

$$\begin{aligned} &1 \leq S(P,1) \leq PI(P) \leq P \\ &1/P \leq E(P,1) \leq U(P) \leq 1 \\ &1 \leq R(P,1) \leq T/E(P,1) \leq P \\ &1/P \leq E(P,1) \leq C(P,1) \leq 1 \\ &Q(P,1) \leq S(P,1) \leq PI(P) \leq P \end{aligned}$$

PI is an upper bound for S, and U is an upper bound for E, with equality iff O(P)=O(1) so that R=C=1. The standard of comparison, an optimal serial computation, has PI, S, U, E, R, C and Q all equal to unity.

The last relationship above shows four successively refined measures of the performance improvement of parallel versus serial processing. First, P=min(p,P') indicates the maximum processor bandwidth, or the maximum speed of the parallel computation. Then, the Parallel Index indicates the average processor bandwidth, or average speed, of the computation. Third, the Speedup indicates the effective processor bandwidth, or effective speed, of the computation. Finally, the Quality is a single performance measure that takes into account mainly the speed improvement, but also the efficiency and the redundancy of parallel versus serial processing.

TABLE 1: OPTIMIZATION OF PERFORMANCE MEASURES AND TOP-FORM PARAMETERS

Performance Measure	TOP-form parameter
(1) maximizing Speedup	= minimizing T(P)
(2) maximizing Efficiency	= minimizing PxT(P)
(3) maximizing Compression (minimizing Redundancy)	= minimizing O(P)
(4) maximizing Quality (minimizing each component of TOP-form with emphasis on time)	= all of the above: minimizing $T^2.O.P$

Measures of Central Tendency. To characterize the performance of a set of computations rather than an individual computation on a given parallel organization, measures of the central tendency of the data are desired. In table 2, the sample mean of the performance measures and TOP-form parameters are given, and in table 3, the median values are given. In table 4, another measure of central

tendency is introduced, called the aggregate performance measures [9]. The aggregate performance measures are performance measures defined for the aggregate computation, which is the computation consisting of every step of every computation in the set of computations. In other words, the aggregate computation is the end-to-end concatenation in time of all the computations in the set. In parallel-serial comparisons, the aggregate performance measure is a ratio of the sum of the requirements of all the serial computations in the sample, divided by the sum of the corresponding requirements of all the equivalent parallel computations. For example, the aggregate speed measures for a set of computations are defined as:

$$S^a = \sum T(1) / \sum T(P) = \overline{T(1)} / \overline{T(P)}$$

$$PI^a = \sum O(P) / \sum T(P) = \overline{O(P)} / \overline{T(P)}$$

PI^a and S^a may be regarded as the average and effective parallel execution bandwidths, respectively, for a set of computations, in a single program environment, i.e., when the execution of the next computation in the set does not start till the execution of the current computation has ended. Since not all the processors are utilized by a computation during all steps of its execution, the introduction of multiprogramming could reduce the overall execution-time of all the computations in the set, though it cannot reduce further the execution time of any individual computation. Hence, PI^a and S^a may be interpreted as lower bounds for the average and effective parallel execution bandwidths in a multiprogramming environment.

Whereas the mean performance measures give equal weight to each computation in the set, the aggregate performance measures tend to weigh each computation by the relative magnitudes of its computation-size and execution-time. It seems reasonable that the overall performance should be more affected by a longer computation than by a shorter one. In general, the aggregate performance measures indicate the performance of all computations considered as a whole, whereas the mean performance measures indicate the expected performance for any one computation in the set of computations.

3 The Experiments

The raw data is obtained from runs of the Illinois Analyser version 2 [7], which transforms ordinary serial programs into equivalent parallel computations. The Analyser incorporates sophisticated algorithms for recognising serial program constructs and converting these to parallel program constructs for fast and efficient parallel execution. Version 2 (1978) of the Analyser differs from version 1 (1973) [5] mainly in the improved handling of linear recurrences [2] found in the serial program.

Existing Fortran programs (ANSI standard) were obtained from various locations, like the Air Force Weapons Laboratory, Burroughs Corporation, the collected algorithms published by the ACM, a well-known scientific library of programs called EISPACK, some of the old programs from the Illinois Analyser Version 1 (prior 1973), and other miscellaneous sources. These programs are run through the Illinois Analyser version 2, which produces as output, the dependency graph of each program. This is then entered as input to simulators, which restructure the computation when necessary, to execute on either an unlimited MIMD organization, an unlimited SIMD organization, or a p-limited SIMD organization where $p=2^i$, for $i=1,2,\ldots,14$. Hence, 16 different parallel processor organizations are compared with the uniprocessor organization.

Various data on the nature of the serial program and its parallel equivalent are collected, from which we abstract sixteen sets of raw TOP-forms and the raw serial computation size, O(1), for each computation, to use as inputs to our analysis programs. First, a standardization procedure is performed, to ensure that only acceptable parallel-serial comparisons of performance are produced. Essentially, the standardization consists of estimating the optimal serial computation size and the optimal parallel TOP-form for each computation-architecture combination [9]. The performance measures are then calculated from these standardized TOP-forms. The rows labelled "SIMDB" and "MIMDB" in tables 2,3 and 4 refer to the unlimited SIMD and MIMD cases, respectively. The row labelled "M/S ratio" gives the statistics for the ratio of values in the unlimited MIMD over the unlimited SIMD cases, to compare the effect of the added control restriction of SIMD parallel architectures. All the statistics in the tables are calculated for the entire sample of 355 computations.

4 The Results

TOP-form parameters: When the same sample of 355 computations is executed with varying degrees of limited physical parallelism, both the mean and median execution-times decrease, and both the mean and median computation-sizes increase, as the number of processors increases. In each case, the sample mean is about two orders of magnitude larger than the sample median, indicating a distribution that is skewed to the right. The mean and median execution-times in an unlimited MIMD environment are about 60% of the corresponding execution-times in an unlimited SIMD environment.

The median values of the maximum number of processors required, P', indicate that if up to 64 parallel processors are available in the architecture, more than half the computations executed will utilize all the processors available during execution. For this sample of

TABLE 2: MEAN PERFORMANCE MEASURES AND TOP-FORM PARAMETERS

P	PI	S	U	E	R	Q	T	O	P'
1	1.00	1.00	1.000	1.000	1.00	1.00	47524.1	47524.1	1.0
2	1.46	1.36	0.732	0.682	1.08	0.95	28926.9	49427.4	1.7
4	2.53	2.18	0.633	0.546	1.20	1.36	14817.8	49745.2	3.3
8	4.31	3.49	0.539	0.436	1.35	2.02	8538.1	50819.3	6.5
16	7.19	5.57	0.449	0.348	1.52	3.07	5372.4	51359.2	12.4
32	11.73	8.73	0.366	0.273	1.72	4.69	3664.6	54437.8	23.0
64	19.04	13.47	0.297	0.210	2.51	7.14	2700.8	56381.2	42.0
128	28.72	20.10	0.224	0.157	3.62	10.45	2224.1	67413.5	72.2
256	44.85	29.48	0.175	0.115	5.87	15.07	2019.9	91305.2	119.1
512	68.83	44.73	0.134	0.087	6.59	23.22	1883.5	96505.7	199.7
1024	98.07	63.84	0.096	0.062	8.31	31.00	1803.6	120378.9	314.7
2048	136.27	89.12	0.067	0.044	8.50	40.60	1768.4	120494.6	480.5
4096	172.87	116.74	0.042	0.029	8.90	44.95	1754.6	121472.6	693.7
8192	220.70	136.51	0.027	0.017	11.91	37.75	1747.4	140873.1	1134.4
16384	282.62	180.96	0.017	0.011	16.64	51.11	1741.7	156339.9	1748.9
SIMDB	1792.65	187.68	0.309	0.261	145.20	118.53	1620.5	294950.9	235922.1
MIMDB	2524.83	364.01	0.248	0.199	145.23	168.12	1058.0	295011.7	236718.0
M/S	2.07	2.01	0.975	0.974	1.03	2.14	0.6	1.0	4.1

TABLE 3: MEDIAN PERFORMANCE MEASURES AND TOP-FORM PARAMETERS

P	PI	S	U	E	R	Q	T	O	P'
1	1.00	1.00	1.000	1.000	1.00	1.00	607.0	607.0	1
2	1.49	1.24	0.745	0.622	1.00	0.65	553.5	673.0	2
4	2.74	2.01	0.686	0.502	1.00	0.83	335.0	866.5	4
8	4.50	3.04	0.562	0.380	1.02	0.78	254.0	991.0	8
16	6.64	4.24	0.415	0.265	1.03	0.72	162.5	1005.5	16
32	8.57	4.80	0.268	0.150	1.03	0.49	132.5	1062.0	32
64	10.45	5.60	0.163	0.087	1.06	0.31	110.0	1217.0	64
128	11.50	5.79	0.090	0.045	1.07	0.19	87.0	1210.0	94
256	12.77	5.81	0.050	0.023	1.08	0.11	78.0	1198.5	100
512	13.80	6.11	0.027	0.012	1.09	0.06	69.0	1262.5	100
1024	14.06	6.11	0.014	0.006	1.09	0.03	66.5	1372.5	100
2048	13.96	6.27	0.007	0.003	1.08	0.01	65.0	1372.5	100
4096	13.96	6.36	0.003	0.002	1.08	0.01	63.0	1372.5	100
8192	14.36	6.36	0.002	0.001	1.09	0.00	61.0	1402.5	100
16384	14.36	6.76	0.001	0.000	1.09	0.00	61.0	1416.5	100
SIMDB	16.12	7.57	0.205	0.153	1.12	0.99	60.5	1451.0	100
MIMDB	29.16	12.65	0.164	0.117	1.13	0.95	32.0	1451.0	208
M/S	1.58	1.58	1.000	1.000	1.00	1.33	0.6	1.0	2

FIG. 2: PARALLEL INDEX vs. p

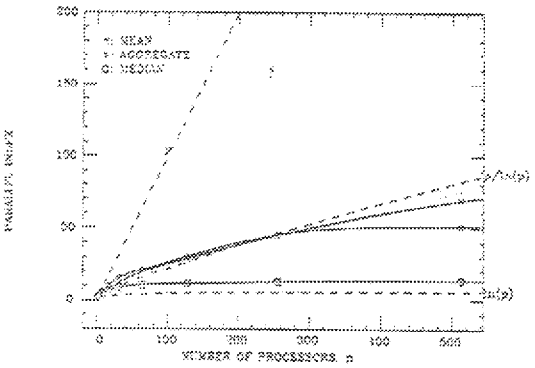


TABLE 4: AGGREGATE PERFORMANCE MEASURES

P	PI	S	U	E	R	Q
1	1.00	1.00	1.000	1.000	1.00	1.00
2	1.71	1.64	0.854	0.821	1.04	1.59
4	3.36	3.21	0.839	0.802	1.05	3.41
8	5.95	5.57	0.744	0.696	1.07	6.06
16	9.56	8.85	0.598	0.553	1.08	9.77
32	14.85	12.97	0.464	0.405	1.15	13.33
64	20.88	17.60	0.326	0.275	1.19	18.89
128	30.31	21.37	0.237	0.167	1.42	20.96
256	45.20	23.53	0.177	0.092	1.92	10.79
512	51.24	25.23	0.100	0.049	2.03	5.61
1024	66.74	26.35	0.065	0.026	2.53	2.79
2048	68.14	26.87	0.033	0.013	2.53	1.42
4096	69.23	27.09	0.017	0.007	2.56	0.71
8192	80.62	27.20	0.010	0.003	2.96	0.34
16384	89.76	27.29	0.005	0.002	3.29	0.17
SIMDB	182.01	29.33	0.004	0.001	6.21	1.19
MIMDB	278.83	44.92	0.005	0.001	6.21	1.47
M/S	1.53	1.53	1.198	1.197	1.00	1.24

FIG. 1: A PARALLEL COMPUTATION

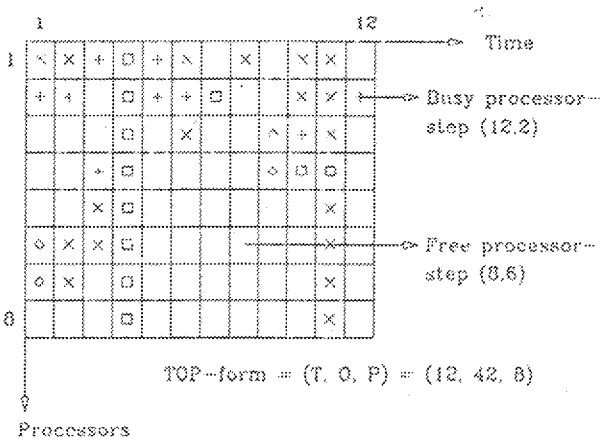


FIG. 3: SPEEDUP vs. p

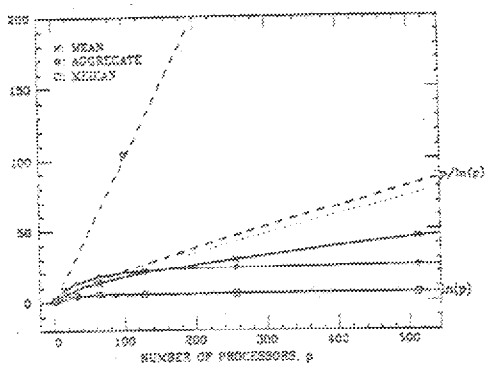


FIG. 4: MIMD SPEED VS. P'

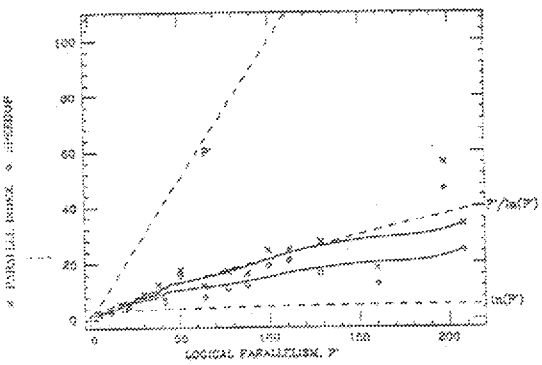


FIG. 5: UTILIZATION vs. p

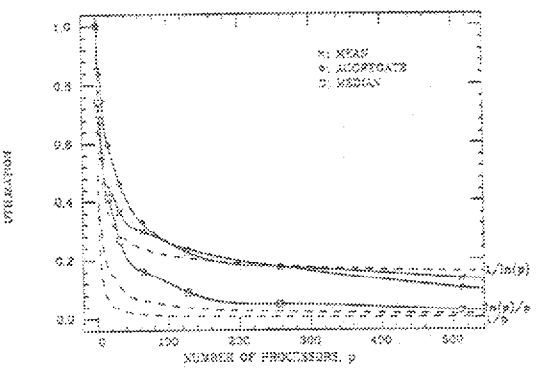


FIG. 6: EFFICIENCY vs. p

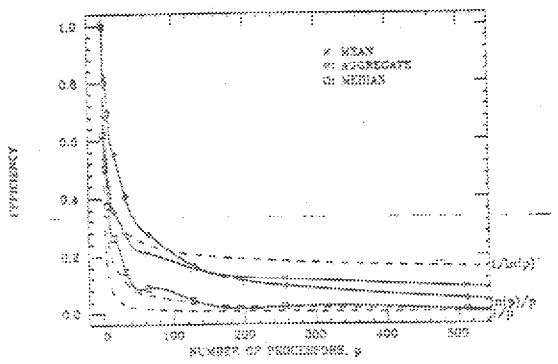


FIG. 7: MIMD UTILIZATION & EFFICIENCY vs. P'

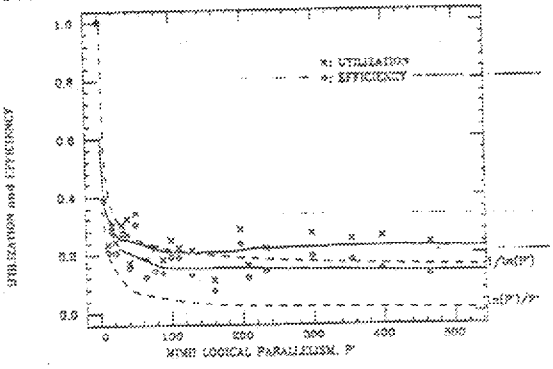


FIG. 8: REDUNDANCY vs. p

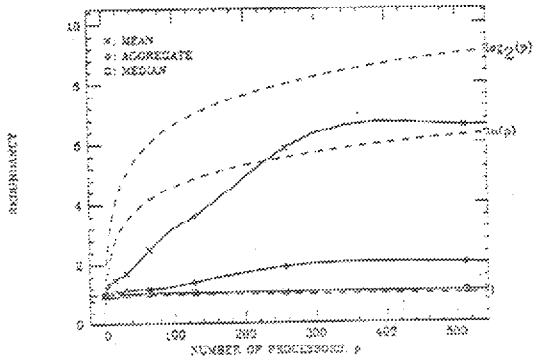


FIG. 9: MIMD AND SIMD REDUNDANCY vs. P'

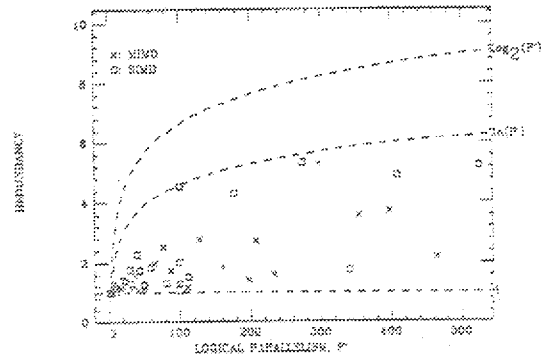
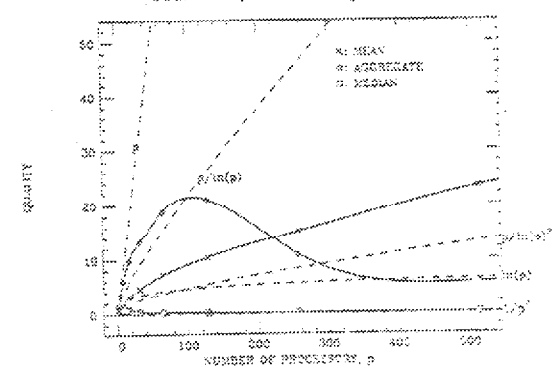


FIG. 10: QUALITY vs. p



computations executing under an SIMD environment, half the computations require not more than 100 parallel processors. Also, half the computations use twice as many processors when an MIMD organization is assumed than when an SIMD option is assumed.

Parallel Index and Speedup. In [9], probabilistic hypotheses on the parallelism distribution in computations were proposed, yielding simple characterizations of the average speed, PI. Necessary and sufficient conditions were given for PI to have a value on the order of P/ln(P), and for PI to have an upper bound of P/ln(P). The natural logarithm function of P, ln(P), is used as an approximation to the Pth. Harmonic number, H(P). The predicted values of PI, which are upper bounds for S in all acceptable parallel-serial comparisons, agree well with empirical observations.

In figure 2, the average and aggregate values of PI are well approximated by the p/ln(p) curve, for $p \leq 300$. For larger p, the average and aggregate values of PI are less than p/ln(p). Similarly, in figure 3, the average and aggregate values of S are well approximated by p/ln(p), for $p \leq 100$, and less than p/ln(p) for larger p. The ln(p) curve forms a lower bound in each case.

The mean, aggregate and median PI values tend to run approximately parallel to the corresponding Speedup values. Hence, the trend of these measures of central tendency (mean, median, aggregate) of the Speedup values are well predicted by the corresponding trend of the PI values, and vice versa.

Figure 4 plots the Parallel Index and Speedup values, averaged over every ten consecutive points, assuming an MIMD organization with unlimited physical parallelism. The solid lines are the smoothed curves automatically generated for the crosses (PI) and diamonds (S) by the plotting package [1], using a smoothing algorithm involving running means, running medians, quadratic interpolation and "hanning" [11]. There is excellent agreement between the observed average PI values and the P'/ln(P') curve, for this range of P'. The Speedup values tend to lie below the P'/ln(P') curve.

Similar plots for the unlimited SIMD computations indicate that there are no significant differences in the trends of the observed PI and S values when compared with the unlimited MIMD case.

Binomial tests [4, 9] with a significance level of 5% were performed which indicate that the majority (more than 50%) of computations encountered have Parallel Indices and Speedups less than P'/ln(P'), in an SIMD or MIMD environment with unlimited physical parallelism. In fact, for unlimited MIMD computations,

$$\text{Prob}\{S(P',1) < P'/\ln(P')\} > 0.75$$

In an SIMD environment with p-limited physical parallelism, the majority of computations have Parallel Indices and Speedups less than p/H(p) for p sufficiently large ($p \geq 64$ processors for PI, $p \geq 16$ processors for S). More than 80% of the computations have Speedups less than p/H(p), for $p \geq 256$ processors.

Utilization and Efficiency. For any individual computation:

$$PI(P) = U(P).P, \quad \text{and} \quad S(P,1) = E(P,1).P,$$

where P may be interpreted as the physical parallelism, p, in a p-limited parallel architecture, or as the logical parallelism, P', in an unlimited parallel architecture.

This relationship between the Parallel Index and Utilization measures, and between the Speedup and Efficiency measures also holds for the corresponding pairs of mean, median and aggregate values for a set of p-limited computations. For example,

$$\overline{PI(p)} = \overline{U(p,1)} \cdot p, \quad \text{and} \quad \overline{S(p,1)} = \overline{E(p,1)} \cdot p.$$

It is sometimes hypothesized that the Speedup is a linear function of p, of the form, k.p, for some constant $k < 1$. However, figures 5 and 6 clearly show that the mean, aggregate and median values of U and E are not constants independent of p. Hence, it is impossible for the corresponding values of PI and S to be linear functions of p. In fact, the mean, median and aggregate values of U and E are decreasing convex functions of p, implying that the corresponding values of PI and S are increasing concave functions of p. Note that the p/ln(p) characterization of PI and S is an increasing concave function of p.

In the case of unlimited physical parallelism, the smoothed curves of U(P') and E(P',1) are well approximated by the 1/ln(P') curve (figure 7). However, if the first data point is ignored, it is not clear that U(P') and E(P',1) are necessarily decreasing convex functions of P'. In fact, they could be described as very slightly decreasing linear functions of P', which may even be considered constant functions, independent of P'. The first data point plotted has P', U(P') and E(P',1) all identically equal to 1. The ten computations represented by this data point are those where the optimal parallel computation is in fact a serial computation, since the minimax degree of parallelism, P', is equal to 1. Except for this first data point, all the other averaged U and E values lie between 0.1 and 0.4. These empirical observations suggest the following

Hypothesis on the Conservation of Processor-Time. The ratio of the processor-time requirement of an optimal serial computation to that of an equivalent optimal parallel computation is:

$$E(P',1) = \frac{Cmin}{P' \times Tmin} = k, \quad \text{where } k \leq 0.5.$$

This hypotheses on the conservation of processor-time does NOT imply that when more processors are available to execute a given computation, then the execution-time will decrease accordingly, so that the processor-time requirement stays constant. Rather, it implies that the optimal parallel processor-time requirement is more than twice the optimal serial processor-time requirement, and the ratio of the two quantities appears to be fairly constant over many different computations.

There are no major differences in the utilization of the processor-time resource, when the computations are structured for an SIMD organization rather than an MIMD organization, with unlimited physical parallelism.

Redundancy. In the empirical results, the median redundancy is less than 1.15 for all MIMD and SIMD computations, with unlimited and limited degrees of physical parallelism. Hence, half the computations achieve a parallel execution-time less than the serial execution-time, with the introduction of less than 15% of redundant operations compared with the serial computation size. Although the mean redundancy is less robust than the median redundancy to extreme values, it is less than O(log(P)) (figures 8 and 9).

A variable X is said to be positively associated or in agreement, with another variable Y if large values of X tend to occur with large values of Y, and small values of X tend to occur with small values of Y. Similarly, X is said to be negatively associated, or in disagreement, with Y if large values of X occur with small values of Y, and small values of X occur with large values of Y. The Spearman rank correlation coefficient, R, may be used to test the degree and direction of association between any pair of variables. The magnitude of R, $0 \leq |R| \leq 1$ gives the degree of association, and the sign of R gives the type (agreement or disagreement) of association.

From the tests of association based on the Spearman correlation coefficient in both unlimited SIMD and MIMD cases, the redudancy measure is found to be negatively associated with the Efficiency and Quality, positively associated with P' and the Parallel Index, and not associated with the Speedup. It has also been observed [10] that larger redundancies are associated with larger probabilities of numerical instability in the parallel computation, as compared with the serial equivalent. Hence, parallel computations with large redundancies should be avoided, since these tend to be associated with inefficient computations with low qualities and higher probabilities of numerical instability. Also, since the Redundancy measure is found to be independent of the Speedup measure, redundant operations should be introduced into a parallel computation only if this increases the Speedup (effective speed), and not just the Parallel Index (average speed), of the resultant parallel computation.

Quality. In the tests of association based on the Spearman correlation coefficient, the Quality measure is found to be independent of the logical parallelism, P', in both SIMD and MIMD computations, assuming unlimited physical parallelism. This is a desirable result for the chosen definition of the Quality measure, since the quality of processing should not be biased towards computations with either smaller or larger degrees of logical parallelism.

The empirical median quality decreases as the physical parallelism increases beyond p=4, and is less than one, in all cases. Hence, more than half the computations in each p-limited and unlimited SIMD and MIMD case have higher qualities when executed as a serial computation than when executed as a parallel computation.

Unlike the relationship between the mean quality and mean speedup, the definition of the aggregate quality does not constrain it to have an upper bound given by the aggregate speedup measure. In the sample of computations examined, the aggregate Quality increased in value, at approximately the same rate as the aggregate Speedup, up to p around 100 (figure 11). As p increased beyond this "saturation point", the aggregate Speedup began to level off and the aggregate Quality declined. This behaviour is representative of most individual and aggregate computations, and hence the Quality measure may be used to chose the optimal number of processors to use in executing a given computation, or set of computations.

Figure 12 shows the frequency distribution of the values of p at which the highest quality is attained for each computation in the sample. Almost half (46%) of the computations examined attain their highest quality value of one, at p=1 (serial computations). The next largest frequency occurs at p=16 and p=32, where about eight percent, each, of the computations attain their highest quality values. The cumulative relative frequency curve indicates that about ninety percent of the computations attain their highest quality values for p<256. If this sample is representative of computations in general, then the parallel processor organization need not have more than 256 processors, in order that ninety percent of the computations executing on it may attain their highest quality potential. It is an interesting coincidence that the ILLIAC IV, an SIMD machine, was originally designed to have a maximum of 256 parallel processors.

5 Conclusions

The characterization of the performance measures varies according to the maximum degree

of parallelism, P. Hence, we define the following approximate ranges:

Let $P=O(1)$ denote $1 \leq P \leq 5$, and $P=O(10^i)$ denote $5.10^{i-1} \leq P \leq 5.10^i$, for $i=1,2,...$

For p-limited architectures, the best characterization for the mean or aggregate values of the speed measures are (figures 13a,b,c):

Approx. range of p	Speed: PI, S
p = O(1)	O(p)
p = O(10) or O(100)	O(p/ln(p))
p = O(1000)	$O(\ln(p)) \leq PI,S \leq O(p/\ln(p))$
p = O(10000) or more	O(ln(p))

In the case of unlimited physical parallelism, PI and S are best characterized as O(P'/ln(P')), for all P'=O(1000) or less.

These empirical observations support the speed characterization of parallel computations given in [9]:

"For general technical computations, the measures of central tendency such as the mean, median and aggregate values of the Parallel Index and the Speedup, all lie between $k_1.\ln(P)$ and $k_2.P/\ln(P)$, where $0<k_1\leq 1$ and $k_2 \geq 1$. Furthermore, the majority of computations will also have individual PI and S values between these lower and upper bounds. P may be interpreted as either the logical parallelism P', in an environment with unlimited physical parallelism, or as the physical parallelism, p, for sufficiently large p, in an environment with limited physical parallelism. So,

$$\max(1, k_1.\ln(P)) \leq S(P,1) \leq PI(P) \leq \min(k_2.P/\ln(P), P)"$$

Suppose that $k_1=0.5$ and $k_2=2$. Then, for $P \geq 8$, $\ln(P)/2$ is greater than 1, and $2.P/\ln(P)$ is less than P. Hence, except for the smallest P values, the $O(\ln(P))$ and $O(P/\ln(P))$ bounds form increasingly tighter bounds for S and PI, as P increases, when compared with the absolute limits of 1 and P.

The empirical speed characterization may be used as a rough guide to the minimum number of parallel processors needed to attain a certain average or effective parallel execution bandwidth. For example, if an average parallel execution bandwidth of ten operations per time-unit is desired, then at least 36 parallel processors should be used, since $p/\ln(p) = 36/\ln(36) = 10.05$. This predicted speed of $O(p/\ln(p))$ operations per time-unit for p=O(10) processors should be regarded as the expected speed potential, since in practice, interactions between processors, memories and other components of the computer organization will cause further performance degradations. Conversely, given a parallel processor organization with limited physical parallelism of degree p, the appropriate speed characterization given above may be used to estimate its expected speed potential.

For p-limited architectures, the best characterization for the efficiency measures, U and E, are obtained by dividing the corresponding values of PI and S by p. For example, if p=O(10) or O(100), U and E are characterized by $O(1/\ln(p))$. Hence, if an efficiency of at least 25 percent is desired, then less than 60 parallel processors should be used, since $1/\ln(p) = 1/\ln(59) = 0.25$, but $1/\ln(60) = 0.24$. For a smaller efficiency, more parallel processors may be used.

For p-limited architectures, the empirical mean and aggregate Utilization and Efficiency measures substantiate the observation that the corresponding mean and aggregate PI and S measures are increasing concave functions of p, like $p/\ln(p)$, and not increasing linear functions of p.

For unlimited architectures, the averaged Utilization and Efficiency measures defined with respect to the logical parallelism, P', suggested a hypothesis on the conservation of processor-time.

Parallel computations with large Redundancy measures should be avoided since these are associated with inefficient computations with low qualities and higher probabilities of numerical instability. Also, redundant operations should be introduced into parallel computations only if this decreases the parallel execution-time when compared with known equivalent serial execution-times.

The Quality measure may be used to choose the optimal number of processors to use in executing a given computation or set of computations.

The mean, median and aggregate values of the M/S ratios for the Speedup and Quality measures imply that the performance improvement of MIMD versus SIMD organizations is less than two and a half times. This same performance improvement may also be obtained by increasing the number of processors available in the architecture. For example, to upgrade the performance of a given p-limited SIMD architecture, the incremental cost of conversion to the less restrictive MIMD organization should be compared with the incremental cost of adding more parallel processors to the SIMD architecture.

Acknowledgements

I am indebted to Professor David Kuck, Robert Kuhn, Michael Wolfe, Bruce Leasure, Pen-Chung Yew and other colleagues at the University of Illinois, for permission to use the raw data generated by the Illinois Analyser version 2. The invaluable database built up at

Illinois represents probably the largest and most significant experimental effort on the speedup of existing serial programs by parallel processing. This paper represents only one possible study of a small subset of the data available in the Illinois database.

I also wish to thank Professor Michael Flynn of Stanford University for his advice and help.

References

[1] Chaffee, R.,B., "Top Drawer", CGTM No. 178, SLAC Computation Group, Stanford, California, revised August 1978.

[2] Chen,S.C., and Kuck,D., "Time and Parallel Processor Bounds for Linear Recurrence Systems", IEEE Transactions on Computers, c-24 no. 7, July 1975, pp.701-717.

[3] Flynn,M., "Some Computer Organizations and Their Effectiveness", IEEE Transactions on Computers, c-21, Sept 1972, pp.948-962.

[4] Gibbons, J. D., "Nonparametric Statistical Inference", McGraw Hill, 1971.

[5] Kuck,D., Budnik,P., Chen,S.C., Davis,E., Han,J., Kraska,P., Lawrie,D., Muraoka,Y., Strebendt,R., and Towle,R., "Measurements of Parallelism in Ordinary Fortran Programs", 1973 Sagamore Computer Conference on Parallel Processing.

[6] Kuck, D.J., "A Survey of Parallel Machine Organization and Programming", Computing Surveys, vol. 9 no. 1, March 1977, pp. 29-58.

[7] Kuck,D., Kuhn,R., Wolfe,M., Yew,P.,and Leasure,B., "A Database of Parallel Programs Automatically Generated from Ordinary Fortran Programs", private communication, October 1978.

[8] Leasure, B., "Compiling Serial Languages for Parallel Machines", Report No. 805, Department of Computer Science, University of Illinois at Champaign-Urbana, November 1976.

[9] Lee, R.B., "Performance Characterization of Parallel Processor Organizations", Ph.D. thesis, Stanford University, May 1980, distributed by University Microfilms International, 300 North Zeeb Road, Ann Arbor, Michigan 48106.

[10] Sameh,A., "Numerical Parallel Algorithms - A Survey", Proceedings of the Symposium on High Speed Computer and Algorithm Organization, April, 1977, (invited), Academic Press pub. 1977, (D. Kuck, D. Lawrie, A. Sameh, eds.).

[11] Tukey, J., "Exploratory Data Analysis", Addison-Wesley Publishing House, 1977.

FIG. 11: AGGREGATE SPEEDUP & QUALITY vs. p

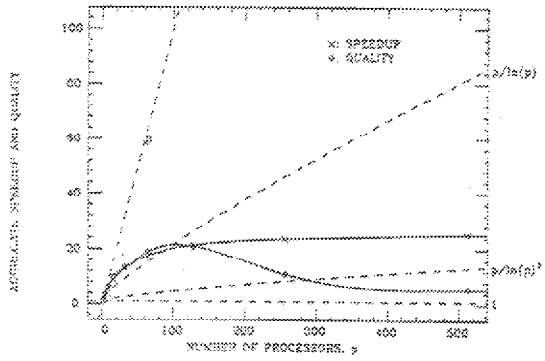


FIG. 12: PROCESSORS USED FOR HIGHEST QUALITY

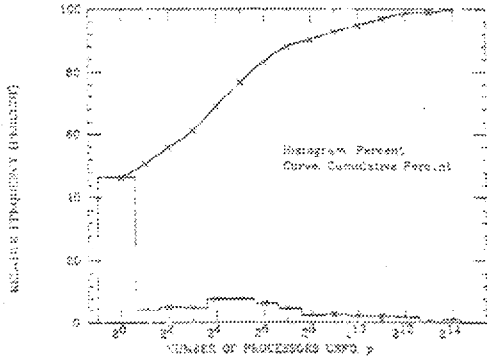


FIG. 13: MEAN PI AND SPEEDUP vs. p