Reprinted from:
PERSPECTIVES ON COMPUTER SCIENCE: From the 10th Anniversary Symposium at the Computer Science Department, Carnegie-Mellon University
© 1977
ACADEMIC PRESS, INC.
NEW YORK SAN FRANCISCO LONDON

Multiplicative Speedup of Systems

R. Reddy and Allen Newell

Department of Computer Science
Carnegie-Mellon University
Pittsburgh, Pennsylvania

This paper attempts to show that speedups of several orders of magnitude are possible in systems where a hierarchy of levels exists. Typical of these are large complex computer systems such as speech- and image-understanding systems. There are many levels of potential improvement in such systems. The levels range from device technology, computer architecture, and software architecture at the lower levels to system organization, algorithmic analysis, knowledge sources, and heuristics at the higher. Potential speedups at each of these levels can be estimated through a careful study of several specific examples that have resulted in speedup. Speedup of components combines additively in systems which are the usual assemblage of components. However, in hierarchical structures where the levels are relatively independent, improvements at various levels combine multiplicatively, resulting in very large total speedup.

1. INTRODUCTION

The problem of speedup of programs and systems is of interest to most of us. At Carnegie-Mellon University we have been attempting to characterize the nature of potential speedup in systems for some time. While we are all aware of some of the common tricks of the trade, such as reprogramming in assembly language, it appears that most of us are unwilling or unable to examine the total range of options that are available to us. In this chapter we will (re)introduce the notion of a hierarchy of levels where improvements at various levels combine multiplicatively. We propose a hypothesis that speedup of several orders of magnitude is possible in systems with many levels of hierarchy.

Our speedup hypothesis is based on several assumptions:

(1) The system has a hierarchy of levels amenable to optimization.

(2) The levels are independent in that optimization at one level does not preclude optimization at the other levels.

(3) The system has not already been optimized at some of the levels.

(4) The problem is important, i.e., many people are working on it and instant speedup is not expected (several years of effort may be required to achieve the potential speedup).

The viewpoint presented here is obviously speculative. However, we hope to present evidence from some current experiments, illustrating the type of speedup that has been achieved at various levels, in support of our hypothesis. We will use examples mostly from local work, where we understand the numbers. Equivalent examples and numbers could no doubt be drawn from many other sources. Indeed, our hypothesis suggests that the evidence for it should be widespread.

The problem of estimating potential system speedup is motivated by our efforts to develop speech- and image-understanding systems. These systems tend to be complex, computationally expensive, and run very slowly on present-day computer systems. Thus it becomes essential to study and understand the expected performance of eventual systems. Tables I and II attempt to illustrate the magnitude of the problem. Table I gives the processing requirements (in millions of instructions executed to process a second of speech) of several connected-speech recognition systems [Reddy 76]. Typically these systems tend to have less than 250 word vocabularies with an average branching factor of less than 10 words at each choice point. As the size of vocabulary increases to a few thousand words, it is expected that processing power on the order of 100 million instructions per second (mips) may be required for real-time recognition unless better

TABLE I

Computational Power Required by Various Systems to Process a Second of Speech[a]

System	No. of instructions/sec of speech (in millions)	
HEARSAY-I	3–10	
DRAGON	15–60	
LINCOLN	45	Avg. branching factor < 10
IBM-Raleigh	35	
HARPY	10	

[a]From Reddy [76].

TABLE II

Processor Speed Required for an Image Understanding System for Different Size Pictures Assuming 1000 Operations per Picture Element

Desired response time (sec)	Picture $1024 \times 1024 \times 3$ high-resolution TV (mips)[a]	Size $2500 \times 3300 \times 4$ satellite photo (mips)
1	1000	10000
10	100	1000
100	10 ~C.mmp)	100

[a]mips, millions of instructions per second.

heuristics and algorithms are devised to reduce search. This implies that 2 to 3 orders of magnitude speedup over a PDP–10 (KA–10) will be necessary if we are ever to have real-time response from the system.

Table II gives the processing requirements for an image-understanding system. Different systems seem to require anywhere from 500 to 5000 operations per pixel (picture element) for processes such as smoothing, differencing, histogram calculation, correlation, color mapping, or distortion calculation. The Table shows that processing power of 10 to 10,000 mips is needed, depending on the size of the picture and the response time desired. Given that a PDP–10 (KA–10) is about a 0.4 mips processor, we can see that it will be impossible to undertake a major effort in image understanding research on KA–10 class machines. We have noted that the minimal point in the table corresponds approximately to C.mmp (equivalently, a CDC–7600).

The hierarchical structure we see in speech- and image-understanding systems starts at the bottom with the technology, and proceeds upward through architecture, system software, program organization, algorithm analysis, program implementation, knowledge sources, and heuristics. We will first take up each of these levels, working bottom up, identifying the potential for speedup and, where we have numbers, trying to estimate increases in speed that might be obtained. After this, we will return to a discussion of the several assumptions of our speedup hypothesis.

2. TECHNOLOGY

Several significant improvements are expected in the areas of device technology and memory technology in the near future. However, when it comes to estimating potential speedup factors, there are interdependencies

TABLE III

Speed Factors Associated with Different Logic Families

	Gate delay (nsec)	Register times (nsec)	Clock times (nsec)
TTL	10	30	120
Schottley-TTL	3	25	100
MECL 10k	2	10	30
MECL 100k	0.7	3 (?)	10 (?)
Josephson J.	0.15		

among device technologies, memory technologies, and the memory bandwidth; i.e., even if we could make an infinitely fast processor, it cannot run faster than the rate at which data are available from memory. This in turn depends on the architectural decisions about the processor-memory-switch structure. Assuming that these decisions are compatible with overall throughput objectives, we can estimate potential speedup factors that can be expected from technology.

Table III presents some typical speeds of different logic families. Present commercially available systems run at about 100 nsec clock speeds. Systems being designed at present (and expected on the market in a year or two) will run between 10 and 30 nsec clock speeds. The present push to increase the component densities is expected to lead to devices capable of running at about 0.15 to 0.2 nsec gate delays and at about 2 to 5 nsec clock time. This implies about 20 to 50 times speedup over technologies used in present systems or about 5 to 10 over the next generation systems.

The high-speed random-access memories will be based on the same technologies as the processor, with many levels of memory hierarchy. Electron beam memories and video disks will probably satisfy most of the bulk memory requirements. Capacities of 10^9 to 10^{12} with 20 to 500 megabit bandwidths can be achieved using these bulk memory technologies. What is important is the interconnection of these levels in the memory hierarchy so that none of them becomes a bottleneck. This in turn is dependent on the task, the algorithms, and the data migration strategies adopted.

3. ARCHITECTURE

Unlike the case of technology, it is much more difficult to estimate the speedup factors that can be achieved through functional specialization of a processor architecture. If one system exhibits significant speedup over

another, and we wish to compare their architectures, we must first normalize for the differences in the relative speed of technologies and for relative costs (as a measure of complexity and/or number of components used in each architecture). Finally, we can compute the total system speedup knowing the speedup factor for the part that is specialized.

We have several functionally specialized systems in the CMU environment. The graphics processor [Rosen 73, Kriz 73] is designed to interpret tree-structured display lists, and most of the control is embedded in the display list representation. The Xerox Graphic Printer [Reddy *et al.* 72] provides real-time generation of 1600 bit scan line data from characters and vectors through a combination of specialized hardware and software. The SPS–41, a processor designed by Signal Processing Systems Co. for performing fast Fourier transforms, performs a "butterfly multiply" (4 multiplications and 6 additions) in 1 μsec.

The HARP, a 30 nsec programmable processor element currently under development at CMU [Kriz *et al.* 76], indicates speedup factors of 35 to 70 over a PDP–11/40 in executing some expensive speech and vision algorithms. It has a 64 register instruction memory, 64 register data memory, and an 8K $\times$ 16–bit 30 nsec buffer memory; it is attached to the PDP–11 UNIBUS. Using a 3 instruction pipeline execution, and independent paths to instruction and register memories, it achieves a microinstruction execution rate of 30 million/sec.

TABLE IV

Architectural Features of Three Processors Used in Speech and Vision Research[a]

	PDP-11/40	SPS-41	HARP
Gate delay (nsec)	10	5	2
Clock times (nsec)	140	100	30
Instruction time (nsec)	> 2000	200	30
Number of registers	8	256 +	128
Pipeline execution	No	Yes	Yes
Cost	$10K	$30K	$20K
Production volume	large	small	few
Technology normalization factor	1.0	0.5	0.2

[a] In spite of the cost differences, these systems are of the same order of complexity. Thus, the technology normalization factor is based purely on gate delay.

Table IV gives the architectural features of three systems used in the CMU environment for speech and vision research: PDP–11/40, SPS–41, and HARP. Table V gives the comparative performance of these three systems in the execution of some basic procedures used often in speech and vision. The instruction density shows the relative power of SPS–41 and HARP instruction sets when compared with the PDP–11 instruction set. Note that these microprogrammable processors require only 2 to 3 instructions on the average to equal the power of a PDP–11 instruction. The code density, measured by the total number of bits required to represent the same algorithm, is another measure of the power of an instruction set. The time of execution is given in part C. Finally, in part D we normalize the execution time first with respect to the PDP–11/40 as unity, and then by a factor based on technology. The PDP–11/40 uses TTL logic (gate delay of 10 nsec). The SPS–41, using mixed TTL and Schottky-TTL (gate delay 5 nsec), has a technology advantage factor (TAF) of 2. HARP, using MECL 10k (clock time 30 nsec), has a TAF of 5. With a speedup 45 to 55 (except in data-limited cases such as the histogram program given as the third example of Table V), the HARP architecture seems to provide a normalized speedup factor of about 10 over a general-purpose computer architecture,

TABLE V

Comparison of Three Architectures: Speedup Resulting from Functionally Specialized Architecture

Program	PDP-11/40	SPS-41	HARP
A. Code size: number of instructions/program			
1. Autocorrelation	36	78	55
2. Dragon loop	35	65	64
3. Histogram	10	21	42
B. Code density: number of bits/program			
4. Autocorrelation	832	1954	880
2. Dragon loop	976	1495	1024
3. Histogram	240	483	672
C. Execution time/program in microseconds			
1. Autocorrelation	90.4	15.6	1.65
2. Dragon loop	89.5	13.0	1.92
3. Histogram	20.9	4.2	1.26
D. Speedup from functional specialization[a]			
1. Autocorrelation	1	5.8 (2.9)	54.8 (11.0)
2. Dragon loop	1	6.9 (3.5)	46.6 (9.3)
3. Histogram	1	5.0 (2.5)	16.6 (8.3)

[a]Technology normalized speedup in parentheses.