Computer System Capacity Fundamentals[*]

May 3, 1974

D. J. Kuck
Dept. of
Computer Science
University of Illinois

[*]This work was supported by the National Bureau of Standards.

1. Introduction

On mips and mops
and megaflops,
and binary
capacity.

This report is an attempt to outline a formal structure for the study of computer capacity. Several traditional measures will be discussed and some new measures will be introduced. Our goals for the use of measures of computer capacity include:

1) Quantification of upper bounds on a given machine's raw theoretical speed for various kinds of computations.
2) Comparisons between diverse computer systems for some set of computations.
3) Evaluation of the actual performance of a given machine on some job mix compared with its theoretical capacity.
4) Guidelines for improving a given system's cost/performance.

Traditionally, people have often quoted computer speeds in mips (millions of instructions per second). But the execution of an "instruction" yields rather different effects on various machines. The range is from some simple indexing operation on a traditional machine to a vector inner product instruction on a modern pipeline processor. Thus, as computer organizations diverged from one another mops (millions of operations per second) became a more reasonable measure. But, in many numerical calculations, floating-point arithmetic operations are the raison d'etre for the computer and logical operations, shifts, etc., are "overhead". Thus, megaflops (millions of floating-point operations per second)may be the important measure.

Quoting megaflops is of course quite irrelevant for most computations performed in the real world every day. In many computations, e.g., data base management, file processing, simulation, etc., almost no floating-point arithmetic is performed. The primary memory speed and often input/output speeds are the most important to quote in evaluating or comparing machines. Our formulation will include consideration of the type of computation being performed in terms of ratios such as primary memory to processor bandwidth used by a computation.

We will attempt to bring together in a uniform way measures of the speeds of various parts of a computer as well as memory size. The two main measures which concern us are speed (of processor, primary and secondary memory) and size of primary memory. By definition, speeds are given in units per second and bits/second is the simplest such measure. It is traditional to call the bit rate of a communication channel its capacity. Similarly, sizes of memories in bits may be thought of as capacities. Since we shall be discussing speeds and sizes together, it seems reasonable to refer to the whole notion as "computer capacity".

In addition to the above machine characteristics, our model will include characteristics of the programs being executed. In particular, we are concerned with the fractions of a computation which use each of the three major parts of a system: processor, primary memory and secondary memory. Thus, our model could be used by independently measuring machine and program characteristics, and relating them through the capacity surfaces we derive.

One difficult question is how to deal with the control unit. It has the potential to allow the several major parts of a computer to operate simultaneously and thereby increase capacity in a major way. We shall briefly discuss "serial" control whereby only one function can be performed at a time.

Our major attention will be given to computer systems in which the processor, primary and secondary memory all can operate simultaneously in an overlapped way. The models we discuss can be thought of as assuming a perfect "lookahead" control unit. Alternatively, any idleness due to the control unit may be considered to be lumped together with the processor. Degradations in system capacity due to variously constrained control units could be an interesting area for further study. In fact, the control unit could be treated as a fourth dimension in Figure 6 of Section 4.

2. Capacity in Overlapped Machines

In this section we define processor, memory and system capacity. These definitions are given in terms of machine parameters (our α's) and program parameters (our β's). There is a good deal of symmetry in much of the following, and we illustrate this by displaying a number of equations.

Let us consider a clocked machine with a processor, i.e., an arithmetic and logical unit, operating at maximum bandwidth (i.e., data rate) B_p bits/second. Let the primary memory bandwidth be B_m bits/second. We define

$$\alpha_{pm} = \frac{B_m}{B_p} > 0 \qquad \text{and } \alpha_{mp} = \frac{B_p}{B_m} > 0 \ .$$

For any given computation, the total available bandwidth of the processor or memory may not be used. Thus, we define $B_p^u \leq B_p$ to be the bandwidth of the processor which is actually used in some computation. Similarly, we define $B_m^u \leq B_m$ as the used bandwidth of the memory for a given computation. Also, for any given computation we define

$$\beta_{pm} = \frac{B_p^u + B_m^u}{B_p^u} \geq 1 \ ,$$

and

$$\beta_{mp} = \frac{B_m^u + B_p^u}{B_m^u} \geq 1 \ ,$$

so it follows that

$$\beta_{pm} - 1 = \frac{B_m^u}{B_p^u} \geq 0 \ ,$$

and

$$\beta_{mp} - 1 = \frac{B_p^u}{B_m^u} \geq 0 \ .$$

We may interpret $1/\beta_{pm}$ as the fraction of some computation in which the processor is engaged. Similarly, $\frac{1}{\beta_{mp}} = 1 - \frac{1}{\beta_{pm}}$ is the fraction of a given computation in which the memory is engaged.

If we assume that each memory cycle and each processor operation require the same amount of time, then the above can be interpreted as follows. For a machine with a control unit which overlaps memory and processor operation, $1/\beta_{pm}$ is the processor fraction of the total number of instructions executed or the processor fraction of the total bandwidth used for some computation. For a machine with a control unit which allows no overlap of processor and memory operation, $1/\beta_{pm}$ is the processor fraction of the total number of instructions executed. Obviously, similar statements hold for $1/\beta_{mp}$.

Next we consider the notion of the capacity of the processor, the memory and the combination of the two. We shall define capacities in bits/

second. Since we are interested in maximum possible data rates, we shall assume that either the memory or the processor bandwidth is saturated in any given computation we discuss. Thus, all our discussions of capacity will assume that for the type of computation under consideration no faster data rate is possible on the machine we are considering.

Let us define

$$\gamma_m = \frac{B_m}{B_m^u} \geq 1 ,$$

and

$$\gamma_p = \frac{B_p}{B_p^u} \geq 1 ,$$

which we call the <u>memory freedom</u> and <u>processor freedom</u>, respectively. When $\gamma_m = 1$, the computation is said to be <u>memory bound</u> and when $\gamma_p = 1$, the computation is said to be <u>processor bound</u>. As outlined in the preceding paragraph, our subsequent discussions of capacity will assume that either $\gamma_m = 1$ or $\gamma_p = 1$, or both.

We can relate machine parameters (α's), program parameters (β's) and freedoms (γ's) as follows. Since

$$\frac{\gamma_m}{\gamma_p} = \frac{B_m \, B_p^u}{B_p \, B_m^u} = \alpha_{pm} \frac{B_p^u}{B_m^u}$$

and since

$$\frac{\beta_{mp}}{\beta_{pm}} = \frac{B_p^u}{B_m^u}$$

we have

$$\frac{\gamma_m}{\gamma_p} = \alpha_{pm} \frac{\beta_{mp}}{\beta_{pm}} .$$

Since $\alpha_{pm} = 1/\alpha_{mp}$, we can derive a similar expression by interchanging m's and p's in this equation.

Now we define, for any given computation on any given machine with overlapped processor and memory, the <u>processor capacity</u>

$$C_p = \begin{cases} \frac{\alpha_{pm}}{\beta_{pm}-1} B_p & \text{if } \alpha_{pm} \leq \beta_{pm} - 1 \\ B_p & \text{otherwise.} \end{cases} \tag{1}$$

Note that $\alpha_{pm} \geq \beta_{pm} - 1$ is equivalent to

$$\frac{B_m}{B_p} \geq \frac{B_m^u}{B_p^u}$$

so

$$\frac{B_m}{B_m^u} \geq \frac{B_p}{B_p^u}$$

or

$$\gamma_m \geq \gamma_p .$$

But since we are assuming that either $\gamma_m = 1$ or $\gamma_p = 1$, this implies that $\gamma_p = 1$. Thus, in the processor bound situation our definition sets $C_p = B_p$ which is the maximum processor data rate.

On the other hand, if $\alpha_{pm} \leq \beta_{pm} - 1$, it follows that $\gamma_m \leq \gamma_p$, but since $\gamma_m = 1$ or $\gamma_p = 1$ we conclude that $\gamma_m = 1$, and we are memory bound.

Thus, $B_m = B_m^u$. Now the definition of C_p can be rewritten in this case as

$$C_p = \frac{\alpha_{pm} B_p}{\beta_{pm} - 1} = \frac{B_m}{B_m^u / B_p^u} = B_p^u \gamma_m .$$

But since $\gamma_m = 1$, we have $C_p = B_p^u$ in the case of a memory bound computation. Thus, the processor capacity is defined to be the fraction of the processor bandwidth which can be used for this computation, given the fact that memory bandwidth is saturated.

If we rewrite processor capacity as

$$C_p = \frac{\gamma_m}{\gamma_p} B_p ,$$

we can interpret it as B_p if memory freedom is greater than processor freedom for some computations and as B_p times the ratio of the freedoms otherwise. We emphasize that the processor only reaches its maximum capacity B_p when $\gamma_m \geq \gamma_p$.

We can derive an expression for memory capacity C_m with analogous characteristics to processor capacity. Thus, we write

$$C_m = \begin{cases} \dfrac{\alpha_{mp} B_m}{\beta_{mp} - 1} & \text{if } \alpha_{mp} \leq \beta_{mp} - 1 \\ B_m & \text{otherwise.} \end{cases} \tag{2}$$

Since $\alpha_{mp} B_m = B_p$, $\beta_{mp} - 1 = \frac{1}{\beta_{pm} - 1}$ and $B_m = \alpha_{pm} B_p$ we can express C_m in terms of B_p as follows

$$C_m = \begin{cases} (\beta_{pm}-1)B_p & \text{if } \beta_{pm} - 1 \leq \alpha_{pm} \\ \alpha_{pm} B_p & \text{otherwise.} \end{cases} \tag{3}$$

If we define system capacity C_s to be the total system bandwidth available for any calculation, by properly adding Equations 1 and 3 we obtain

$$C_s = \begin{cases} (1+\frac{1}{\beta_{pm}-1})\alpha_{pm} B_p & \text{if } \alpha_{pm} \leq \beta_{pm} - 1 \\ (1+\beta_{pm}-1)B_p & \text{otherwise,} \end{cases}$$

so

$$C_s = \begin{cases} \frac{\alpha_{pm}\beta_{pm}}{\beta_{pm}-1} B_p & \text{if } \alpha_{pm} \leq \beta_{pm} - 1 \\ \beta_{pm} B_p & \text{otherwise.} \end{cases} \tag{4}$$

This can be expressed in terms of B_m as

$$C_s = \begin{cases} \frac{\alpha_{mp}\beta_{mp}}{\beta_{mp}-1} B_m & \text{if } \alpha_{mp} \leq \beta_{mp} - 1 \\ \beta_{mp} B_m & \text{otherwise.} \end{cases} \tag{5}$$

Note that maximum system capacity occurs when both the memory and processor are bound, i.e., $\gamma_p = \gamma_m = 1$. Thus, from Equation 4, if $\alpha_{pm} = \beta_{pm} - 1$ we have

$$\begin{aligned} C_s &= \frac{\alpha_{pm}\beta_{pm}}{\beta_{pm}-1} B_p = (1+\frac{1}{\beta_{pm}-1})\alpha_{pm} B_p \\ &= (1+\frac{1}{\alpha_{pm}})B_m = (1+\frac{B_p}{B_m})B_m = B_m + B_p \; . \end{aligned}$$

Thus, the maximum system capacity is the sum of the maximum processor and memory bandwidths.

To make matters concrete, we give in Figure 1 examples of capacities for $\alpha_{pm} = 2$ and various β_{pm} values. In Figure 1 we denote activity by X and inactivity by 0. We show two columns under the label "memory" to denote that the memory bandwidth is twice the processor bandwidth, i.e., $\alpha_{pm} = 2$. The capacities are shown under the columns of activity. Overall results are plotted in Figure 2.

In Figure 3 we plot system and processor capacity for various values of α_{pm}. Note that the processor can perform at its maximum capacity over a wider range of problems (β_{pm} values) for larger α_{pm}. Note also that the memory capacity which is available for memory to memory (or I/O) operations becomes greater for larger α. It should be remarked that as β_{pm} approaches 1, reasonable system performance depends on a high frequency of register to register operations (or cache to cache operations).

$\beta_{pm} = 4/3$

processor	memory	
X	X	0
X	0	0
X	0	0
X	X	0
X	0	0
X	0	0
.	.	
.	.	
.	.	

$$B \quad + \quad B/3 \quad = \frac{4B}{3}$$

$\beta_{pm} = 3/2$

processor	memory	
X	X	0
X	0	0
X	X	0
X	0	0
.	.	
.	.	
.	.	

$$B \quad + \quad \frac{B}{2} \quad = \quad \frac{3B}{2}$$

$\beta_{pm} = 2$

processor	memory	
X	X	0
X	X	0
X	X	0
.	.	
.	.	
.	.	

$$B \quad + \quad B \quad = \quad 2B$$

$\beta_{pm} = 3$

processor	memory	
X	X	X
X	X	X
X	X	X
.	.	
.	.	
.	.	

$$B \quad + \quad 2B \quad = 3B$$

Figure 1. Overlapped Processor and Memory, $\alpha_{pm} = 2$

$\beta_{pm} = 4$

processor	memory
0	X X
0	X X
X	X X
X	X X
X	X X
X	X X
0	X X
0	X X
X	X X
X	X X
X	X X
X	X X
.	.
.	.
.	.

$$\frac{2B}{3} + 2B = \frac{8B}{3}$$

$\beta_{pm} = 5$

processor	memory
0	X X
X	X X
0	X X
X	X X
0	X X
X	X X
.	.
.	.
.	.

$$\frac{B}{2} + 2B = \frac{5B}{2}$$

Figure 1 (continued). Overlapped Processor and Memory, $\alpha_{pm} = 2$

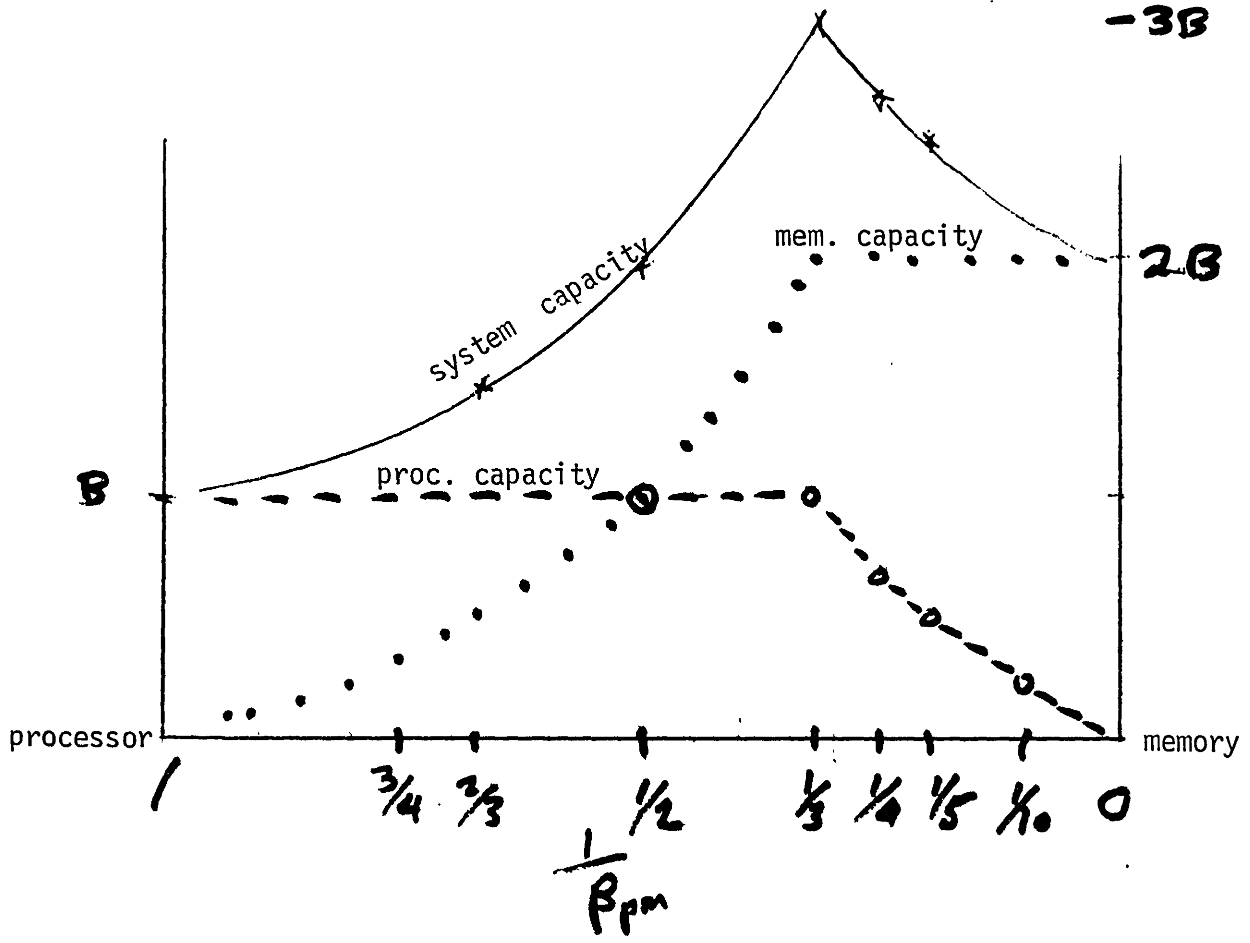

Figure 2. Capacity for $\alpha_{pm} = 2$

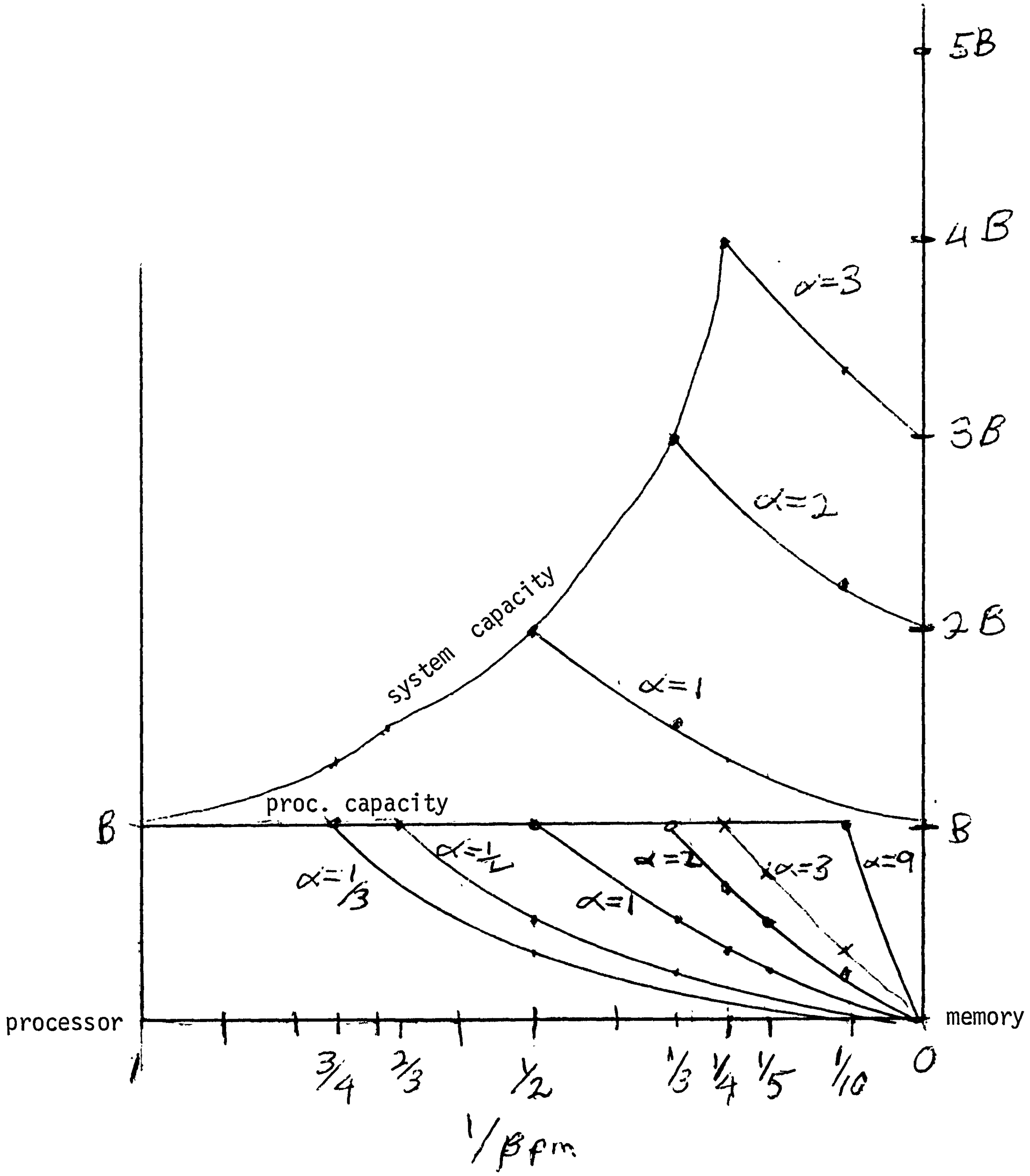

Figure 3. Capacities for Various α_{pm} Values

3. Capacity in Non-Overlapped Machines

To contrast the previous section with a simpler machine and demonstrate how capacities vary as a function of machine organization, we now disallow the simultaneous operation of memory and processor. However, we do assume a perfect lookahead control unit. Figure 4 illustrates the situation for $\alpha = 2$.

It may be seen that in the case of non-overlapped processor and memory, we have (using the notation of the previous section):

$$C_p = \frac{\alpha_{pm} B_p}{\alpha_{pm} + \beta_{pm} - 1} , \qquad (6)$$

$$C_m = \frac{\alpha_{pm}(\beta_{pm} - 1) B_p}{\alpha_{pm} + \beta_{pm} - 1} \qquad (7)$$

and

$$C_s = \frac{\alpha_{pm} \beta_{pm} B_p}{\alpha_{pm} + \beta_{pm} - 1} \qquad (8)$$

We plot the capacity of a non-overlapped machine for $\alpha_{pm} = 2$ in Figure 5. Note the contrast with Figure 2, an overlapped machine. Here the processor and memory capacities only reach their maximum bandwidth at the limits of $1/\beta_{pm}$. Note also that a good deal less system capacity is left over for I/O activities.

We can easily show that an overlapped machine's capacities are all greater than or equal to a non-overlapped machine's. Thus, from Equations 1 and 6 we see that

$$\text{non-overlapped } C_p = \frac{\alpha_{pm} B_p}{\alpha_{pm} + \beta_{pm} - 1} \leq \left\{ \begin{matrix} \dfrac{\alpha_{pm} B_p}{\beta_{pm} - 1} \\ \\ B_p \end{matrix} \right\} = \text{overlapped } C_p$$

since $\alpha_{pm} > 0$ in the first case, and $\beta_{pm} \geq 1$ in the second case. In similar ways we can show that

non-overlapped $C_m \leq$ overlapped C_m

and

non-overlapped $C_s \leq$ overlapped C_s .

$\beta = 2$

processor	memory
0	X X
X	0 0
X	0 0
0	X X
X	0 0
X	0 0
.	.
.	.
.	.

$$2B/3 + 2B/3 = \frac{4B}{3}$$

$\beta = 3$

processor	memory
0	X X
X	0 0
0	X X
X	0 0
0	X X
X	0 0
.	.
.	.
.	.

$$B/2 + B = \frac{3B}{2}$$

$\beta = 4$

processor	memory
0	X X
0	X X
X	0 0
0	X X
X	0 0
0	X X
0	X X
X	0 0
0	X X
X	0 0
.	.
.	.
.	.

$$\frac{2B}{5} + \frac{6B}{5} = \frac{8}{5}B$$

$\beta = 5$

processor	memory
0	X X
0	X X
X	0 0
0	X X
0	X X
X	0 0
.	.
.	.
.	.

$$\frac{B}{3} + \frac{4B}{3} = \frac{5B}{3}$$

Figure 4. Non-overlapped Processor and Memory, $\alpha = 2$

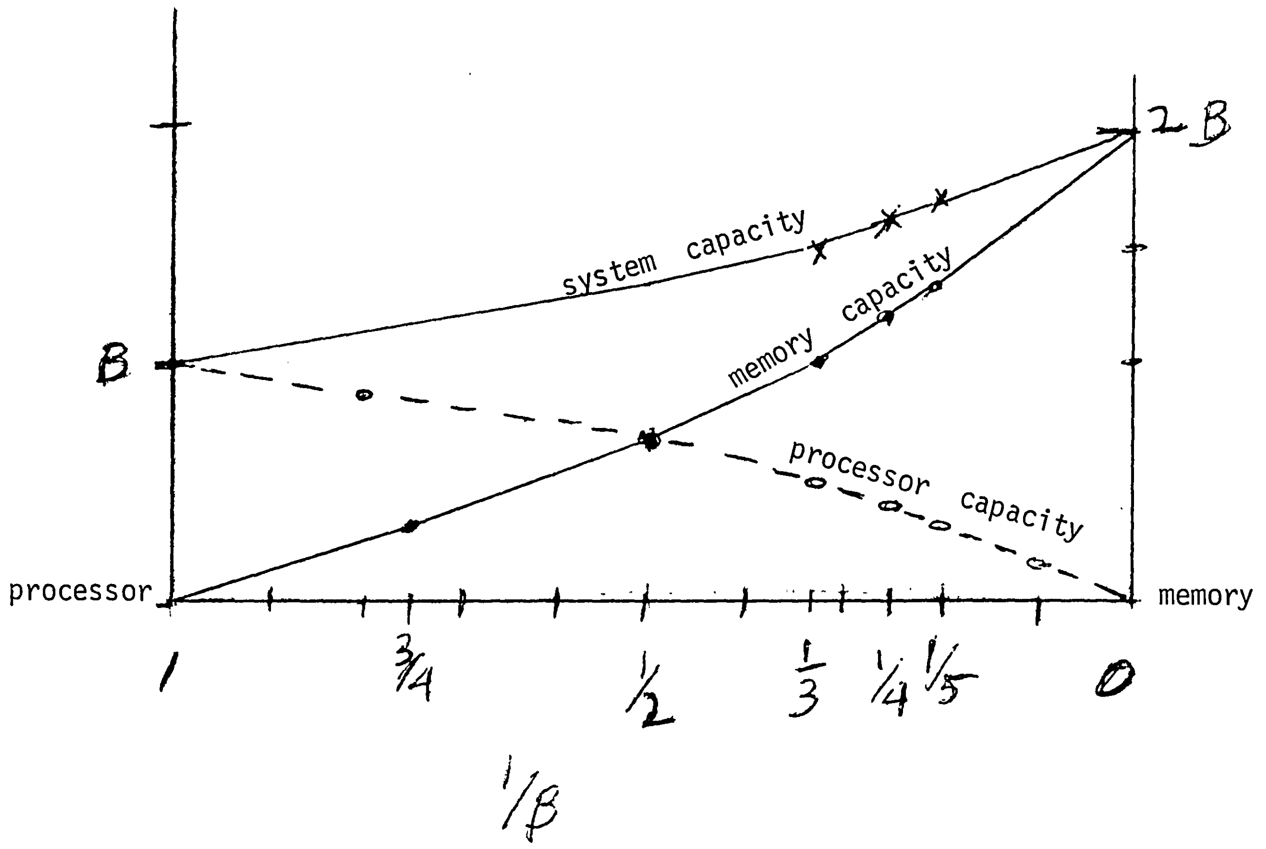

Figure 5. Non-overlapped Capacity for $\alpha = 2$

4. Processor-Memory-Disk Systems

Now we turn to a complete system with three components--processor and primary memory as above, together with a secondary memory which we shall refer to as a disk. We shall assume at all times that one of these three components is operating at its highest data rate, i.e., its bandwidth is saturated. We also assume a control unit which overlaps the operation of the processor, the primary memory and the disk. We first give some definitions which are analogous to those of Section 2.

Let B_d be the disk or I/O bandwidth.

Then

$$\alpha_{pd} = \frac{B_d}{B_p} \quad , \quad \alpha_{md} = \frac{B_d}{B_m}$$

and

$$\alpha_{dp} = \frac{B_p}{B_d} \quad , \quad \alpha_{dm} = \frac{B_m}{B_d} \quad .$$

We also define

$$\beta_{pd} = \frac{B_p^u+B_d^u}{B_p^u} \quad ,$$

$$\beta_{md} = \frac{B_m^u+B_d^u}{B_m^u} \quad ,$$

with β_{dp} and β_{dm} being defined similarly. It follows that processor capacity may be written as:

$$C_p = \begin{cases} \dfrac{\alpha_{pm} B_p}{\beta_{pm}-1} = \dfrac{B_m}{\beta_{pm}-1} & \text{if } \alpha_{pm} \leq \beta_{pm}-1 \ , \\[2ex] \dfrac{\alpha_{pd} B_p}{\beta_{pd}-1} = \dfrac{B_d}{\beta_{pd}-1} & \text{if } \alpha_{pd} \leq \beta_{pd}-1 \ , \\[2ex] B_p & \text{if } \alpha_{pm} \geq \beta_{pm}-1 \text{ and } \alpha_{pd} \geq \beta_{pd}-1 \ . \end{cases}$$

Similarly, we have for memory capacity:

$$C_m = \begin{cases} \dfrac{\alpha_{mp} B_m}{\beta_{mp}-1} = \dfrac{B_p}{\beta_{mp}-1} & \text{if } \alpha_{pm} \geq \beta_{pm}-1 \\[2ex] \dfrac{\alpha_{md} B_m}{\beta_{md}-1} = \dfrac{B_d}{\beta_{md}-1} & \text{if } \alpha_{md} \leq \beta_{md}-1 \\[2ex] B_m & \text{if } \alpha_{pm} \leq \beta_{pm}-1 \text{ and } \alpha_{md} \geq \beta_{md}-1 \ , \end{cases}$$

and for disk capacity:

$$C_d = \begin{cases} \dfrac{\alpha_{dm} B_d}{\beta_{dm}-1} = \dfrac{B_m}{\beta_{dm}-1} & \text{if } \alpha_{md} \geq \beta_{md}-1 \\[2ex] \dfrac{\alpha_{dp} B_d}{\beta_{dp}-1} = \dfrac{B_p}{\beta_{dp}-1} & \text{if } \alpha_{dp} \leq \beta_{dp}-1 \\[2ex] B_d & \text{if } \alpha_{md} \leq \beta_{md}-1 \text{ and } \alpha_{pd} \leq \beta_{pd}-1 \ . \end{cases}$$

By summing these capacities for consistent conditions, we obtain saturated system capacities as follows:

$$C_s = \begin{cases} \left(1 + \frac{1}{\beta_{mp}-1} + \frac{1}{\beta_{dp}-1}\right) B_p & \text{if } \alpha_{pm} \geq \beta_{pm}-1 \text{ and } \alpha_{dp} \leq \beta_{dp}-1 \\ \left(1 + \frac{1}{\beta_{pm}-1} + \frac{1}{\beta_{dm}-1}\right) B_m & \text{if } \alpha_{pm} \leq \beta_{pm}-1 \text{ and } \alpha_{md} \geq \beta_{md}-1 \\ \left(1 + \frac{1}{\beta_{pd}-1} + \frac{1}{\beta_{md}-1}\right) B_d & \text{if } \alpha_{pd} \leq \beta_{pd}-1 \text{ and } \alpha_{md} \leq \beta_{md}-1 \, . \end{cases}$$

It should be noted that in each of these three cases, if the conditions are written as equalities then the maximum capacity is obtained. In each case this reduces to

$$\max C_s = B_p + B_m + B_d \, .$$

To make matters concrete, in Figure 6 we sketch a surface for $B_p = B$, $B_m = 2B$, and $B_d = \frac{B}{2}$. The processor capacity is shown as a plateau of height B which runs off to 0 along the memory-disk axis. The top surface is the system capacity. In the region labelled I, the system is processor bound, and in II and III it is memory and disk bound, respectively. Where these three regions meet, the max $S_c = 3.5B$ point is located.

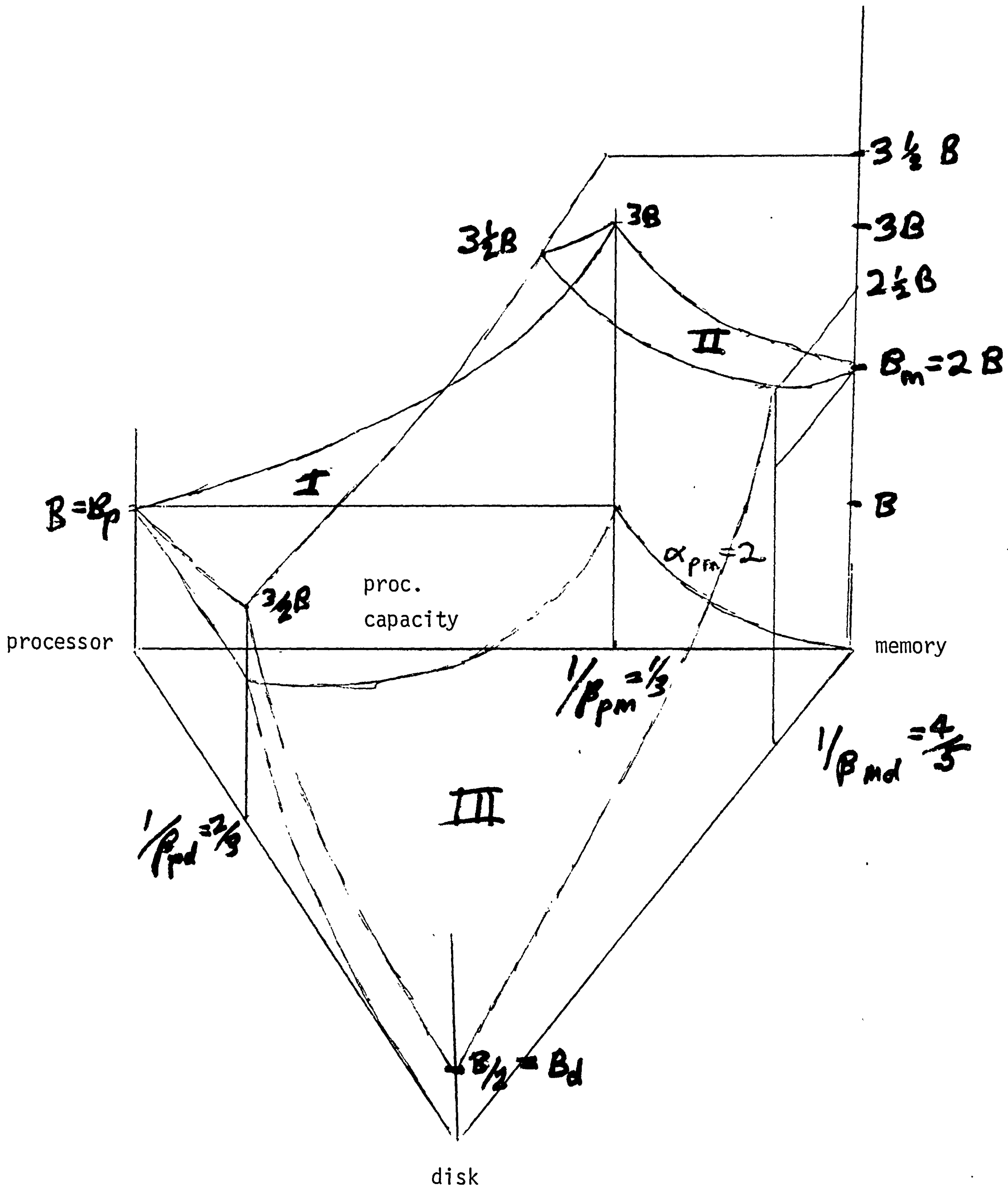

Figure 6. Capacity Space

5. Primary Memory Size vs. B_d

It is well known that there exists a trade-off between primary memory size and I/O bandwidth. Our purpose here is to sketch an analysis of this trade-off and to relate it to our previous discussion of capacity.

Let the primary memory size be N words of w bits each, for a total of wN bits. The time required to fill this memory from a disk of bandwidth B_d (assuming $B_d < B_m$) is wN/B_d sec.

For simplicity, assume a given computation operates on the entire memory. Assume the computation requires N^α time steps. For example, given an $n \times n$ matrix, an n^3 step algorithm would give $\alpha = 3/2$, since $n^2 = N$ if the matrix (or a single $n \times n$ partition) fills primary memory. Now the time required for the entire computation would be wN^α/C_p secs.

On the average, the system would be balanced if the processing time were equal to the input time (assuming no output), that is:

$$wN^\alpha/C_p = wNB_d$$

or

$$N^{\alpha-1} = \frac{C_p}{B_d}$$

which gives us

$$N = \left(\frac{C_p}{B_d}\right)^{\frac{1}{\alpha-1}} \tag{9}$$

as a relationship between memory size N, I/O bandwidth B_d, and processor capacity C_p .

The above model can be easily refined in various ways to provide for input and output of data arrays, to provide for multiple buffering, and so on.

6. Conclusion

The point of this report is to provide a framework for the study of computer capacity. We have explored several aspects of the question and Figure 6 shows a system capacity surface as a function of processor, memory and disk bandwidth. For a given class of computations, this surface corresponds to a memory size given by Equation 9 in Section 5.

While we have glossed over many details, the model described here could be useful in the various ways mentioned in the Introduction.

For example, if we were given a set of computations and a machine configuration we could easily determine a Figure 6 type surface from the machine parameters. From the computational algorithms, we could estimate the various β values as discussed in Section 2. This would allow a determination of our operating point in capacity space. While the ideal point is where $C_s = C_p + C_m + C_d$, a prudent region is probably somewhere between that point and the processor corner of Figure 6 for "numerical" problems. For "business"-type problems it may be between there and the memory corner of Figure 6. For the class of algorithms under consideration, Equation 9 could be used to make memory size trade-offs.

Given some qualitative idea of the operating rules a user prefers, one could use this model to make quantitative sensitivity studies of capacity as a function of bandwidth and memory size. This could lead to improved system cost/effectiveness.

Note that for any given capacity surface, degradation due to operating system overhead, etc., can be quantified by plotting actual performance data in capacity space. In this case, the surfaces shown will serve as theoretical upper bounds on system performance.