A Effect of an Assumption Regarding Communication Time

To aid us in modeling the tradeoff between block bandwidth and block latency, we assume that the processor that has the largest number of words to communicate (C_{max}) also has the largest number of blocks to communicate (B_{max}). This assumption simplifies our calculations by allowing us to pretend that latency-related overheads and bandwidth-related overheads are disjoint parts of each communication phase. In practice our assumption may not be true, and a perfectly accurate model must take into account the fact that one processor's "latency period" may partly overlap another processor's "bandwidth period." However, under such a model it is difficult to give meaningful definitions for "half-bandwidth" and "half-bandwidth latency," so we have chosen to stick with the simplifying assumption that bandwidth and latency are disjoint, even though it causes us to slightly overestimate the length of the communication phase. The question is, how much do we overestimate the required sustained bandwidth as a result?

For a given simulation, this question is easy to answer if we know the machine parameters T_l (block latency) and T_w (inverse block bandwidth). However, we would like to establish a worst-case bound that applies to all machines, regardless of their parameters.

The length of the communication phase is determined by the processor that takes the longest; that is,

$$T_{comm} = \max_{i \text{ is a processor}} (B_i T_l + C_i T_w).$$

Our simplifying assumption is to calculate T_{comm} as if there is some processor i such that $B_i = B_{max}$ and $C_i = C_{max}$. Because the sustained bandwidth we require is $T_c = T_{comm}/C_{max}$, our assumption will cause us to overestimate T_c by no more than

$$\begin{aligned} \beta &= \frac{B_{max} T_l + C_{max} T_w}{\max_i (B_i T_l + C_i T_w)} \\ &= \frac{B_{max} + \alpha C_{max}}{\max_i (B_i + \alpha C_i)} \\ &= \min_i \frac{B_{max} + \alpha C_{max}}{B_i + \alpha C_i}, \end{aligned}$$

where $\alpha = T_w/T_l$ is a parameter representing the ratio between the overhead per word and the overhead per block.

To ensure that our model is reasonably accurate for any machine parameters, we wish to find the value of α that yields the most pessimistic value of β. For argument's sake, suppose there are only three processors, having (B, C) parameters of (B_j, C_j), $(B_{max}, 0)$, and $(0, C_{max})$, respectively (the latter two are the most conservative assumption possible). Because of the last two processors, the denominator $\max_i (B_i + \alpha C_i)$ cannot be less than B_{max}, nor can it be less than αC_{max}; hence, β cannot be greater than two.

Now, consider the first processor j, and the function

$$\beta_j(\alpha) = \frac{B_{max} + \alpha C_{max}}{B_j + \alpha C_j}.$$

It is apparent upon inspection that the derivative $\mathrm{d}\beta_j/\mathrm{d}\alpha$ is either everywhere positive, everywhere negative, or uniformly zero over $\alpha > 0$; hence, there are no critical points, and β_j is maximized at an extreme value of α.

However, the notion of "extreme values of α" is not straightforward, and does not necessarily correspond to the values $\alpha = 0$ and $\alpha = \infty$. Obviously, the processor that maximizes the value of $B_i + \alpha C_i$ depends on α. As α is swept from zero to infinity, the processor (called the *active processor* in the terminology of

optimization) that maximizes this expression may change several times; each α value where such a change occurs determines a candidate for β.

Returning to our three-processor example, we note that when $\alpha = 0$, the active processor is $(B_{max}, 0)$ (because it maximizes $B_j + \alpha C_j$), yielding $\beta = 1$, and when $\alpha \to \infty$, the active processor becomes $(0, C_{max})$, also yielding $\beta = 1$. Hence, the value of α that maximizes β will occur between these extremes. As α increases from zero, the active processor changes from $(B_{max}, 0)$ to (B_i, C_i) when $B_{max} = B_i + \alpha C_i$, and changes from (B_i, C_i) to $(0, C_{max})$ when $B_i + \alpha C_i = \alpha C_{max}$. In each of these two cases, we solve for α and substitute the result into β, yielding

$$\beta\left(\alpha = \frac{B_{max} - B_i}{C_i}\right) = 1 + \frac{C_{max}(B_{max} - B_i)}{C_i B_{max}}$$

and

$$\beta\left(\alpha = \frac{B_i}{C_{max} - C_i}\right) = 1 + \frac{B_{max}(C_{max} - C_i)}{B_i C_{max}}.$$

The larger of these two values is an upper bound on β (over all values of α).

If there are more than three processors, such a bound may be computed for each processor. Each such bound is valid, so we use the smallest. Hence, our model does not overestimate the required sustained bandwidth by more than

$$\beta = 1 + \min_{i \text{ is a processor}} \left[\max \left\{ \frac{C_{max}(B_{max} - B_i)}{C_i B_{max}} \frac{B_{max}(C_{max} - C_i)}{B_i C_{max}} \right\} \right].$$

In general, this bound is not tight. The tightest bound (when α is unknown) may be derived by computing the convex hull of the set of points (B_i, C_i) (where i ranges over all processors), and checking each edge of the convex hull as a possible maximizer of β. (Each edge of the convex hull represents a change in the active processor.)