Homework 2, CS685-002, Spring 2001

Due time: March 21, 2001

This homework can be done in group of at most three students. If you choose to work in group, you must submit the results as one group with all of your names on the submission. The same grade will be assigned to all students in the same group. (**Your submission must be typed.**)

1. (15 points) Let us assume one Jacobi iteration is performed on a hypothetical parallel computer with two processors in a time sharing environment. Let t_r be the time for one Jacobi iteration in one processor without other users, t_0 be the overhead time to establish the parallel interface each time two processors to be used, t_c be the communication time after one iteration. So $T_1 = t_r$ is the run time for one Jacobi iteration in one processor without any other users. $T_2 = t_0 + \frac{t_r}{2} + t_c$ would be the run time in two processors without any other users. In a time sharing environment, one cannot assume the two processors will devote $\frac{t_1}{2}$ time continuously for one Jacobi iteration. Suppose the CPU in each processor is shared with another user, the computation of one Jacobi iteration is finished in n times, each time the elapsed time would be approximately $\tilde{T_i} = t_0 + \frac{t_i}{2}$. The total run time is $T_3 = \sum_{i=1}^{n} \tilde{T_i} + t_c$. What is the condition that performing one iteration in two processors is faster than performing it in one processor, suppose the time is equally shared with another user in both cases? (Hint: establish a bound for the parallel overhead t_0.)

2. (35 points) It is possible to analyze the situation from the point of view of synchronization. Even the Jacobi iteration does not require any communication within each iteration, communications are needed to update the approximate solution in each processor once an iteration is finished and before the next iteration is started. Thus a processor finishing its computation first must wait for other processors to finish their computations. This looks like a barrier function being put at the end of each iteration for synchronization. If the parallel computer is used exclusively by one user, there will be no much problem. However, if there are many users sharing the parallel system, it is very difficult to know which processor will finish its job at what time. This implies that a meaningful speedup will be difficult to achieve if many users are sharing the same parallel system. Try to develop a model, similar to that used in the first problem, to analyze the situation. You model should accommodate more than two processors. It is intuitive that a parallel code will run slower if more processors are used. However, you should give a condition, based on your model, under which the Jacobi iteration will be faster by utilizing more processors. (The obvious condition that the number of other users is zero is not a serious answer.)