CS 590V: Parallel Computing
Spring 1997: Final Exam
May 8, 1997, 10:20 - 12:20 AM

NOTE: This is a closed book, closed notes exam.

1. **Basic Parallel Computing [10 points]** The following statements are true or false? Justify your answer with a two-line statement:

   (a) Locality of reference is only critical for parallel computers and not for serial processors with deep memory (cache) hierarchies.

   (b) Data locality is critical only for message passing machines and not for shared address space machines.

   (c) All communication in a parallel computer can be handled by a single pair of t_s and t_w values.

   (d) Conventional microprocessors increasingly rely on parallelism within the processor for speed.

   (e) Many parallel algorithms (such as matrix multiply) are also good serial algorithms.

2. **Algorithm Design and Analysis [10 points]**

   (a) Consider a problem of adding 256 numbers using 32 threads. Each thread adds 8 numbers. The 32 partial sums are then added to the global sum one after the other. Assuming single cycle addition, zero cost locking, and no communication cost, what is the maximum speedup and efficiency of this code.

   (b) Consider the general problem of adding n numbers on p processors. What is the most efficient way of performing this based on the t_s, t_w model. What is the parallel runtime of your formulation?

3. **Programming in Message Passing [30 points]** Consider the sparse matrix illustrated in Figure 1. The matrix has 8 rows. Consider a matrix with identical structure with n rows. Write a message passing pseudocode for multiplying this matrix with a vector. What is the runtime of this code (assuming time t_c for a single multiply-add and using the t_s, t_w model of communication). Hint: Ordering is important in this example.

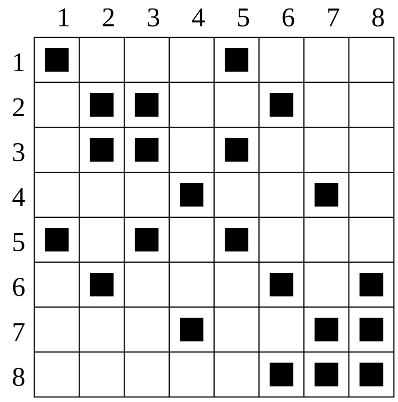

Figure 1: Sparsity structure of the matrix.

4. **Programming in Threads [30 points]** Consider a simple producer-consumer scenario: the producer produces a number (assume that this is returned by a call to the routine `produce()`). The production of the number takes a non-deterministic amount of time. The number is left in a cubbyhole by the producer. The consumer consumes this number using the routine `consume()`. Write a threaded pseudocode for this problem. Use two threads: `producer` and `consumer`. Make sure that the consumer consumes only after producer has produced (i.e. left the number in the cubbyhole). Also, the producer must not overwrite a produced number before it has been consumed.

5. **Programming in HPF [20 points]** Illustrate the distributions induced by the following code fragments:

   (a)
```
REAL DIMENSION(12,6) :: A
!HPF$ TEMPLATE T(12,12)
!HPF$ ALIGN A(I,J) WITH T(2*J-1,I)
```

   (b)
```
REAL DIMENSION(8,8) :: C,D
!HPF$ TEMPLATE T(12,12)
!HPF$ ALIGN C(:,:) WITH T(:,:)
!HPF$ ALIGN D(I,J) WITH T(I+5,J+5)
```

   (c)
```
REAL DIMENSION(4,4) :: B
!HPF$ TEMPLATE T(12,12)
!HPF$ ALIGN B(I,J) WITH T(2:12:3,1:12:3)
```