1. (*15 pts)* Amdahl's Law: Mattson et al. (our book) expresses Amdahl's law as:

$$S(P) = \frac{1}{\gamma + \frac{1-\gamma}{P}}$$

in which γ is the serial fraction of the program, P is the number of processors, and $S(P)$ is the possible speedup.

   (a) What is the maximum possible speedup on 50 processors if 5% of a computation is serial and 95% parallel? You may round to an integer.

   (b) Describe the implications of Amdahl's law; i.e., what does Amdahl's law state in English?

2. (*15 pts*) Synchronized Blocks in Java

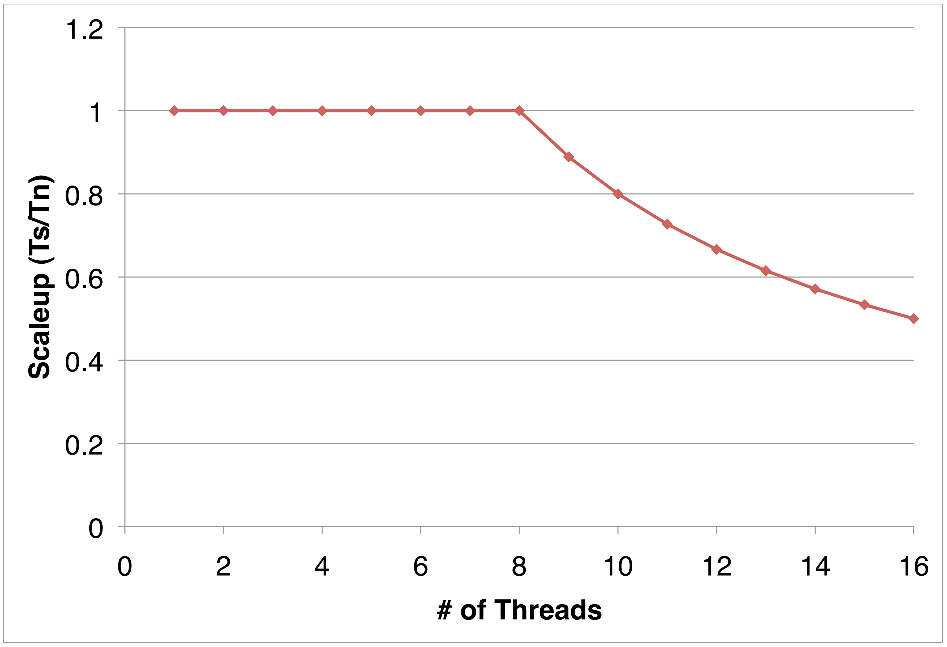
   (a) Make the inner block of the following function *synchronized* with respect to the specific object (self object) in which it is called.

```
public void UpdateSharedState ( .... )
{
   // The following block of code needs to be synchronized


   {
      ....         // bunch of updates to shared variables
   }
}
```

   (b) In part (a), you synchronized on the self object. One may also synchronize on the entire class. **When would one synchronize on the entire class? Why would one prefer to synchronize on a specific object when possible?** (Answer both questions.)

3. (*20 pts*) The following chart displays the practical scaleup realized when running a parallel algorithm on a shared-memory machine with 8 physical cores. The chart shows the relative performance of running a small problem on a single thread versus running a k times larger program on k threads.



(a) What are the scaling properties of *the parallel algorithm*? (That is, how would you expect the algorithm to perform as you increased its parallel resources.)

(b) What practical scaleup was realized on this particular machine? Characterize the data in the chart.

4. (*50 pts*) Map/Reduce for Financial Data The input to the map/reduce program consists of a set of records that represent financial transactions, e.g., the transaction log of a brokerage firm. Records are of the form

$$\text{Input schema: } K = \{\}, V = (\text{symbol}, \text{purchaser}, \text{shares}, \text{price})$$

in which each value record indicates an individual sale of `shares` shares of `symbol` bought by `purchaser` at cost of `price` per share. Analysts at the brokerage firm wish to answer the question

What symbols were bought by the top purchaser?

which will produce outputs of the form

$$\text{Output data: } (\text{purchaser}, \text{symbol})$$

in which a top purchaser does the highest volume (`shares` $\times$ `price`) summed over all of that `purchaser`'s transactions. Write a multi-stage map/reduce program that answers this question. For each stage, you should **(1) specify the input (map) schema and output (reduce) schema** and **(2) describe (in words) what the function does**. **(3) You should also state the cardinality of the inputs and outputs**, e.g. "each map record produces one output record to the reducer" or "the reducer accumulates all records within a key and outputs a single record." The following example gives a sense of how you might answer this question.

Note: You may use the output of a previous map/reduce phase as shared data that is input to all mappers in a subsequent phase. If you do so, that shared data should be small, i.e. one value or one record. Not all solutions require this technique.

Example: To answer the question:

How many shares were bought of each symbol?

One might respond in the following manner.

$$\text{Input Schema: } K = \{\}, V = (\text{symbol}, \text{purchaser}, \text{shares}, \text{price})$$
$$\text{Output Schema: } K = \text{symbol}, V = \text{shares}$$

Map: The mapper performs a key transformation, outputting the `symbol` as the key and the `shares` as the value. The mapper outputs one record for each input record.

Reduce: The reducer sums the `shares` within each `symbol`. It outputs one record for each key that is the `symbol` and the `sum` of the shares.