The goal of Exercise #1 in the presentation is to facilitate a discussion among students about what they think is the best solution and justify their answers. Here are suggested answers to the questions, but they are by no means the only possible answers that students can give.

1. An astronomy model where we want to analyze 16 disjoint galaxies. Calculations will be run on each individual star in each galaxy.

MODEL: Hybrid
WHY: Because each of the 16 galaxies do not require any information from any of the other galaxies, they are an ideal candidate to be run in parallel. Because they can each be run independently, they can use distributed memory and MPI. Depending on the problem size, there can be a large amount of data that needs to be calculated on for each galaxy. This information can be split up and run in parallel as well through shared memory. Because of this, it is an ideal candidate for hybrid parallelism.

2. We want to calculate the photosynthetic rate of each individual leaf of 10,000 different soybean plants. The photosynthetic rate of one plant does not affect the photosynthetic rate of another, but the photosynthetic rate of one leaf of a plant can affect the photosynthetic rate of another leaf on the same plant

MODEL: MPI
WHY: Because each of the 10,000 soybean plants is not contingent on information from other plants, they can be run in a distributed memory fashion. Each leaf of an individual plant is dependent on information from the others on the same plant. While it may seem as though this could then be done in shared memory where each leaf gets run on a thread and the information is shared, there are relatively few leaves on a plant. It is possible that if we try to split a single plant up even further, it would slow down the problem. Because of this, we can simply split the 10,000 plants among MPI processes and run the plants in serial.

3. We need to analyze the amount of traffic that goes through a neighborhood with 4 intersections to determine whether or not putting traffic lights in would be necessary.

MODEL: OpenMP
WHY: There are several ways that this problem can be interpreted, but depending on the interpreted problem, shared memory is most likely the optimal solution regardless. Saying that there are 4 intersections in the neighborhood is more to show how large the problem size is. Because 4 intersections implies a relatively small neighborhood, it does

not make sense to distribute the problem across multiple processes or node and rely on communication. Rather, it makes more sense to simply place the full problem in the same place so that everything has access to everything.

4. We will perform a scalability study on an image processing algorithm to see how well the algorithm scales with really large datasets from material science. The example dataset you will run your strong and weak scalability studies on is 4 TB in size

MODEL: MPI
WHY: The key things to take away from this problem are the fact that we are doing a scalability study, so our problem will need to run on increasingly larger concurrencies, and the fact that we are using a large dataset (4 TB specifically). On most supercomputers or clusters, nodes are not going to have 4 TB of memory per node, so we can rule out running solely in OpenMP and shared memory. While it is possible to run scalabilities studies on hybrid problems, it is extremely difficult. Because of this, we can determine that we will need to use only MPI for this problem and to run the scalability study on an increasing number of MPI processes.

5. The goal is to build a graph that represents a photograph of a landscape as seen from up above (imagine that it is a picture taken from an airplane). The nodes of the graph will represent different sections of the landscape, but each pixel of the graph requires information about other pixels so that it can determine whether or not the pixels are in the same section of the graph that is being generated. Your image you are building the graph of is 512x512 pixels.

MODEL: OpenMP
WHY: An important piece of information to pull out here is that the nodes of the graph require information about other pixels that are represented by other nodes of the graph; i.e., all nodes need information about all other nodes in order to determine whether or not pixels are already represented when generating the graph. The fact that the image is 512x512 means that the problem is relatively small, so the full problem can be placed in shared memory. If the problem size was larger, accommodations would have to be made to be run across multiple nodes and rely on communication, but if the problem is large enough, then the communication would not exceed the computation and the problem could be run in MPI.

6. The visualization we are attempting to build with VisIt contains 27 billion total elements laid out as a cube. There are 3072 files that VisIt will read in. Each one is a sub-grid of the full cube that contains 192x129x256 cells.

MODEL: MPI
WHY: Determining what model to use here actually has nothing to do with the problem size or what we are actually creating a visualization of. In fact, this problem is actually constrained by VisIt itself. While VisIt provides you with the capability to generate visualizations either in serial or in parallel, it only provides you with the ability to use distributed memory and MPI. As of now, VisIt does not support shared memory abilities, so this problem can only be run in just MPI.

7. We want to run 8 completely different chemical reactions and from the data gathered regarding the chemical reactions, we can then build the output molecular model. Two additional pieces of information are gathered for each chemical reaction: the final amount of the reactant as well as the final amount of the product.

MODEL: Hybrid
WHY: This problem is relatively straightforward to see why it can take advantage of a hybrid model. First, the problem tells us that there are 8 completely different chemical reactions, meaning that we can run all 8 of these chemical reactions at the same time. Each reaction does not require any information from any of the other reactions, so it is a perfect candidate to be run across multiple MPI processes. Then, the problem tells us that each chemical reaction outputs 3 completely disjoint pieces of information. This information can be calculated in parallel as well, so it can be run on threads at the same time using OpenMP. Because of this, the problem is an ideal candidate to be run using hybrid parallelism.

Common Pitfalls
As programming for a hybrid distributed memory and shared memory parallelism application can be very difficult due to the required prior knowledge, it can often be a difficult process to grasp and program for. Students who have had no prior programming experience, particularly for high performance computing, might have trouble being able to critically think and decide what programming model would be most beneficial in that instance.