Report:

1. The count variable is a shared variable between the threads because it is declared in the scope of all threads. Each thread accesses the same count variable and sometimes increments it at the same time. This accounts for the inconsistency in the final value.

2. There is no inconsistency with smaller variables because incrementing for a smaller count will be less likely to desynchronize the threads.

3. The local variables are local to each thread. They are declared in the scope of each thread and not shared between each other.

4. The final value will always be consistent because a lock is placed on the count variable every time it is accessed to prevent other threads from accessing it at the same time. This way the variable always increments individually with each thread.

5. The two are so different because the lock/unlock code requires extra operations to lock/unlock. It is substantially slower because of the extra 3 million operations.