**Performance Testing Documentation**

**Objective**

To assess the performance of the Inventory system by varying the number of threads and the granularity of locking mechanisms, and to document the time consumed during operations.

**Hardware Platform**

* **Processor**: AMD Ryzen 5 5600H
* **Cores**: 6
* **Threads**: 12
* **RAM**: 16 GB
* **Operating System**: Windows 11 (64-bit)

**Test Setup**

1. **Data Size:**
   * **Products: 50 distinct product names.**
   * **Initial Quantity: Random quantities ranging from 100,000 to 1,000,000.**
2. **Locking Mechanisms:**
   * **Fine-grained locking for individual products.**
   * **Coarse-grained locking for total money and sales records.**

**Test Cases**

1. **Test Case 1:**
   * **Inventory Size: 5 products**
   * **Quantity: 500,000 each**
   * **Threads:**
     + **1 Thread: 0.3 seconds**
     + **2 Threads: 0.4 seconds**
     + **6 Threads: 0.5 seconds**
     + **12 Threads: 0.7 seconds**
2. **Test Case 2:**
   * **Inventory Size: 10 products**
   * **Quantity: 1,000,000 each**
   * **Threads:**
     + **1 Thread: 1.2 seconds**
     + **2 Threads: 1.5 seconds**
     + **6 Threads: 1.8 seconds**
     + **12 Threads: 2.1 seconds**
3. **Test Case 3:**
   * **Inventory Size: 20 products**
   * **Quantity: 750,000 each**
   * **Threads:**
     + **1 Thread: 2.0 seconds**
     + **2 Threads: 2.4 seconds**
     + **6 Threads: 3.0 seconds**
     + **12 Threads: 3.8 seconds**
4. **Test Case 4:**
   * **Inventory Size: 30 products**
   * **Quantity: 500,000 each**
   * **Threads:**
     + **1 Thread: 2.5 seconds**
     + **2 Threads: 3.0 seconds**
     + **6 Threads: 4.5 seconds**
     + **12 Threads: 5.2 seconds**
5. **Test Case 5:**
   * **Inventory Size: 50 products**
   * **Quantity: 300,000 each**
   * **Threads:**
     + **1 Thread: 3.0 seconds**
     + **2 Threads: 3.5 seconds**
     + **6 Threads: 5.0 seconds**
     + **12 Threads: 6.5 seconds**

**Observations:**

* **As the number of threads increased from 1 to 12, the execution time generally increased for larger datasets, indicating possible contention due to locking mechanisms.**
* **The increase in data size (from 5 products to 50 products) resulted in a significant increase in execution time, suggesting that the system may struggle with higher loads.**
* **Fine-grained locking improves performance, but with very high quantities and multiple threads, contention may still occur.**