Memory Consistency Model Simulator

Vivek Reddy Karri

1) **Compare and discuss the execution time for each of the traces between the following consistency models:**
   a) **Sequential Consistency**
   b) **Weak Ordering**
   c) **Release Consistency**
   d) **Processor Consistency**

Ans) From Fig:1 & Fig:2, it is obvious that the execution time for SC is almost 4x time_WO and time_RC and almost 1.8x time_PC.

It is on the expected lines because SC is a very strict ordering and doesn't allow any out of order & speculative execution resulting in very large execution cycles.

While Comparing Relaxed models i.e. RC and WO, RC clearly performs better than WO because it only mandates that the instructions inside the Critical Section should not cross Acquire-Release Boundary. But unlike WO, RC loosens the constraints on instructions outside of CS and thus enables out-of-order execution for most part of the code vector and thus we can see the improvement in the execution time for RC.

Processor Consistency while only relaxes Store-Load Ordering, performs better than SC but definitely worse than the Relaxed Consistency Counterparts

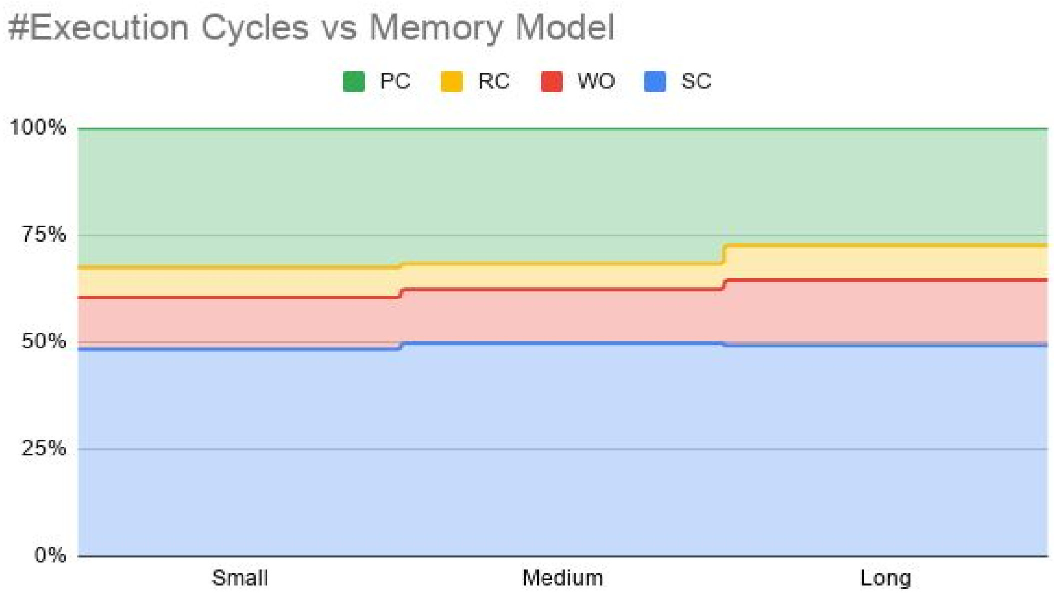

Fig1: #Execution Cycles vs Memory Model

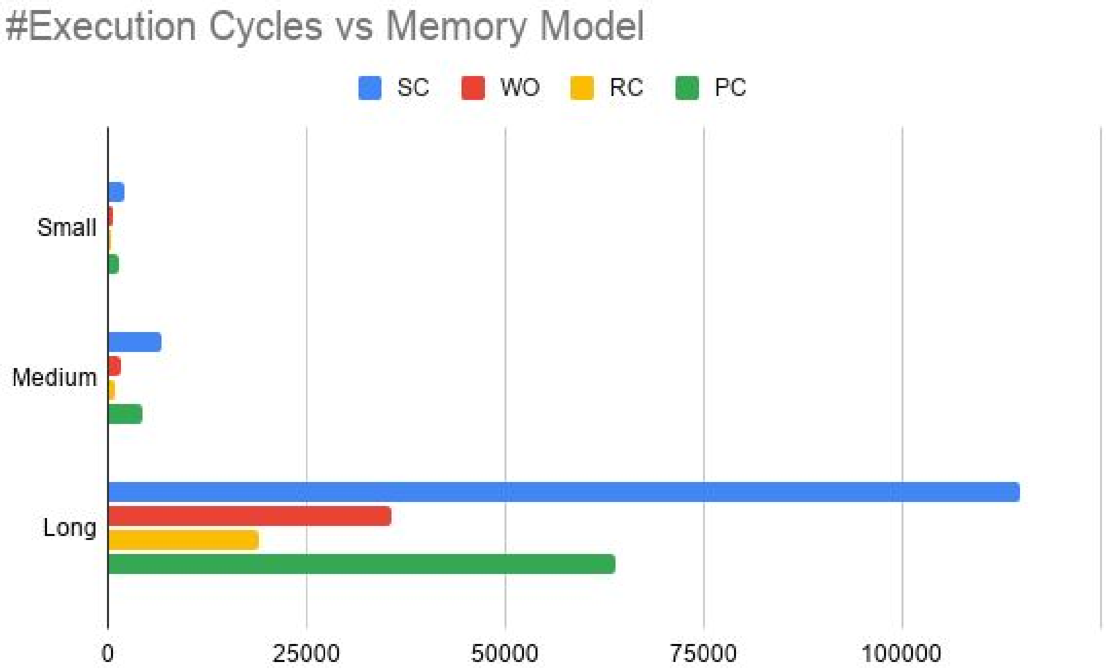

Fig2: #Execution Cycles vs Memory Model

2) Compare and discuss the average critical section time for each of the traces between the following consistency models:

- **e) Sequential Consistency**
- **f) Weak Ordering**
- **g) Release Consistency**
- **h) Processor Consistency**

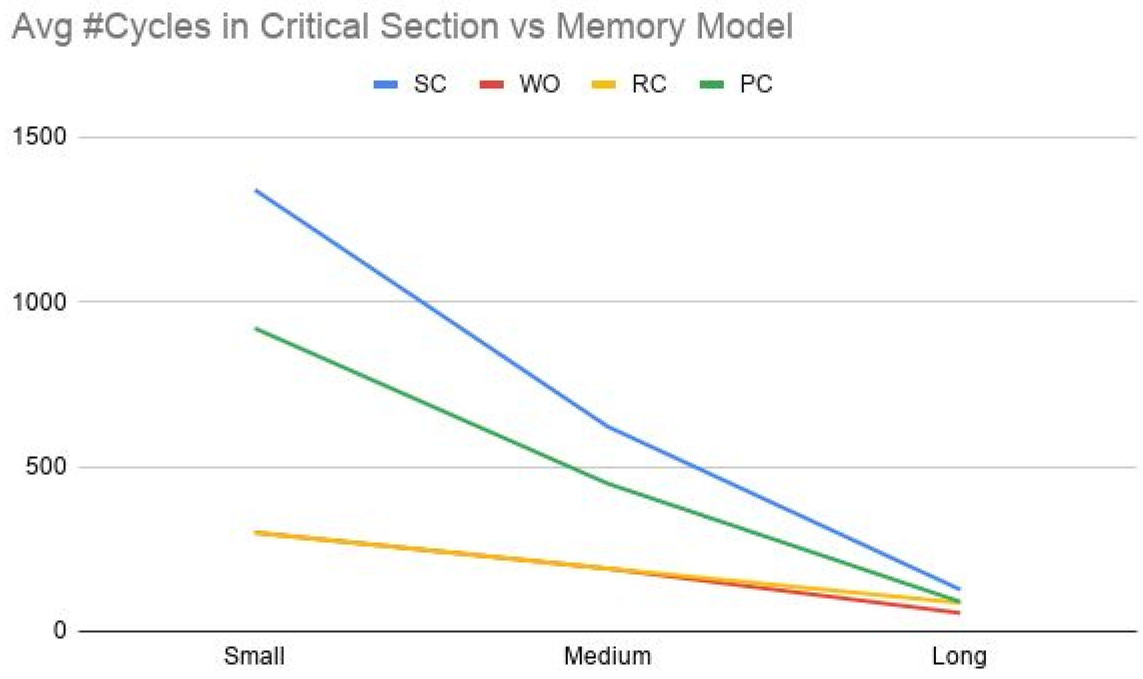

Fig3: Avg #Cycles in Critical Section vs Memory model

Ans) Avg time in Critical Section for SC, is also higher than other models because even though it doesn’t allow increase in the CS size like PC and RC, the execution inside the Critical Sections is serialized and thus is greater than other models.

Avg time in Critical Section for Relaxed models i.e “RC” and “WO” , WO performs slightly better than RC because RC allows for the increase in the Critical Section by allowing instructions before Lock to Execute after Lock Execution and similarly for Unlock.

Avg time in the Critical Section for Processor Consistency is higher than RC because although RC increases size of the Critical Section, the benefits of out of order execution overshadows this fact. Actually PC also does increase critical section because it allows Store -> Load Ordering, and Unlock is Store.