HW4 Conversative Distributed Algorithm

MSIM 406 – Distributed Simulation

Thomas Laverghetta,
Old Dominion University – Computation Modeling and Simulation Engineering

Design

Please see HW4_MSIM406_pseudocode.txt for pseudocode design.

Testing

First test is to see if the simulation will terminate cleanly using just null msgs and lookahead of 5. For this test, I wanted to verify that the simulation will terminate if there are no more non-null msgs in the system. To do this, I initialized the system, set lookahead to 5, then ran the simulation for 20-time-units with 2-, 3-, and 4-processors. The simulation output is below.

Has seen below, the output of the first test was simulation times after null msg execution (each null msg in queue to execute, hence why multiple with same time stamp) and executing process rank (curr=x) where the simulation times stop at 20. This is the correct behavior. Also, this also showed processors not executing events (null msgs) in their past (ascending time).

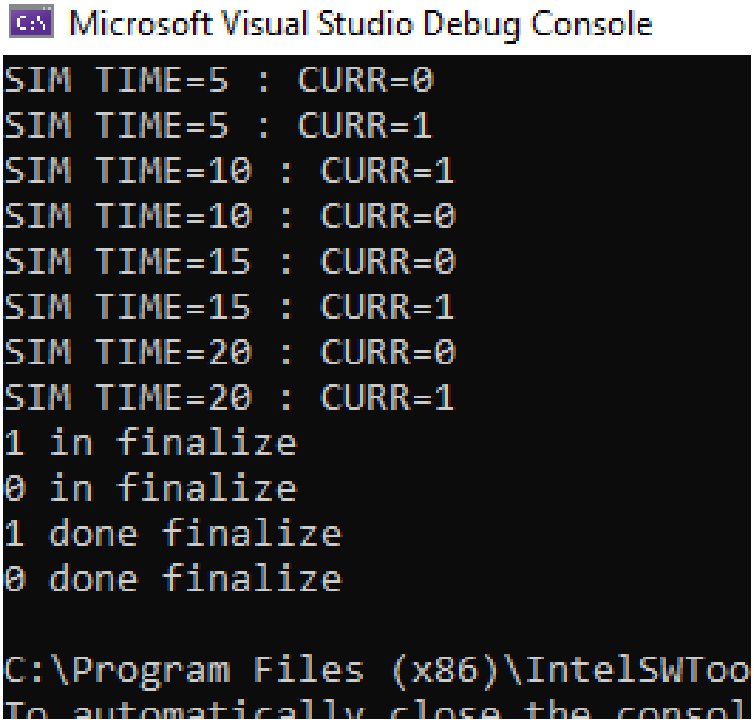

Figure 1. First Test w/2-processors

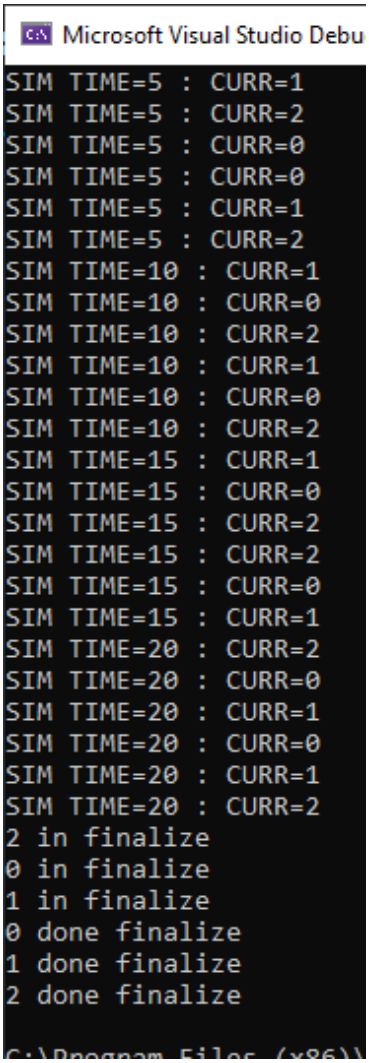

Figure 2. First test with 3 Processors

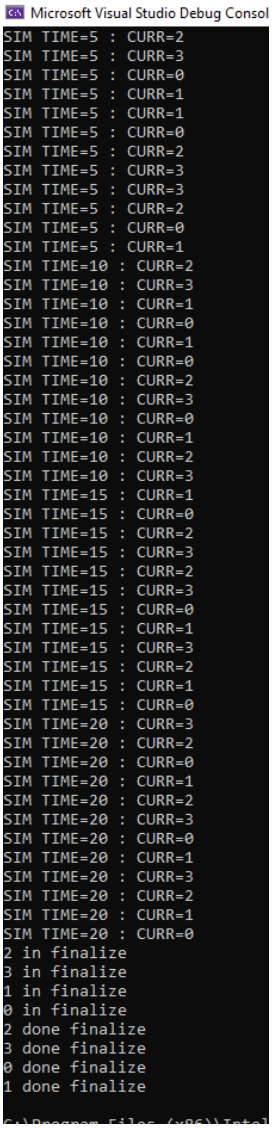

Figure 3. First test w/4 processors

Second test is to verify conservative behavior – i.e., events will never be scheduled in the past. To test this, I created a basic test event action (TestEA) which when execute will schedule event to random process with triangular distribution (5, 15, 25) event time. Thus, by initially scheduling multiple TestEAs, the events will propagate through the system with varying times. Also, to test serialization and deserialization of msg events, TestEA will have two state variables, origin process (where it was created) and ID (event identifier).

For the test, I set lookahead to 5, registered TestEA with simulation executive, initially scheduled w/5-TestEAs per processor, and run simulation for 55-time-units with 2- and 4-processors. The figures below are the outputs. Where each output line is execution of event. The line is formatted with event ID: origin process, curr process rank (CURR=), and simulation time on current process.

Has seen in the figures below, the test was a success! If you follow any single process over the course of the simulation, it never goes back in time (no events scheduled in the past).

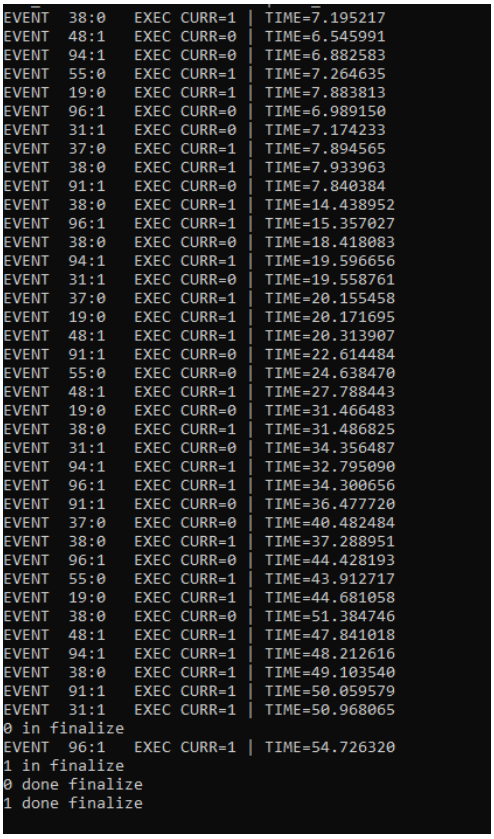

Figure 4. 2nd Test w/2-processors and 5-TestEA/process

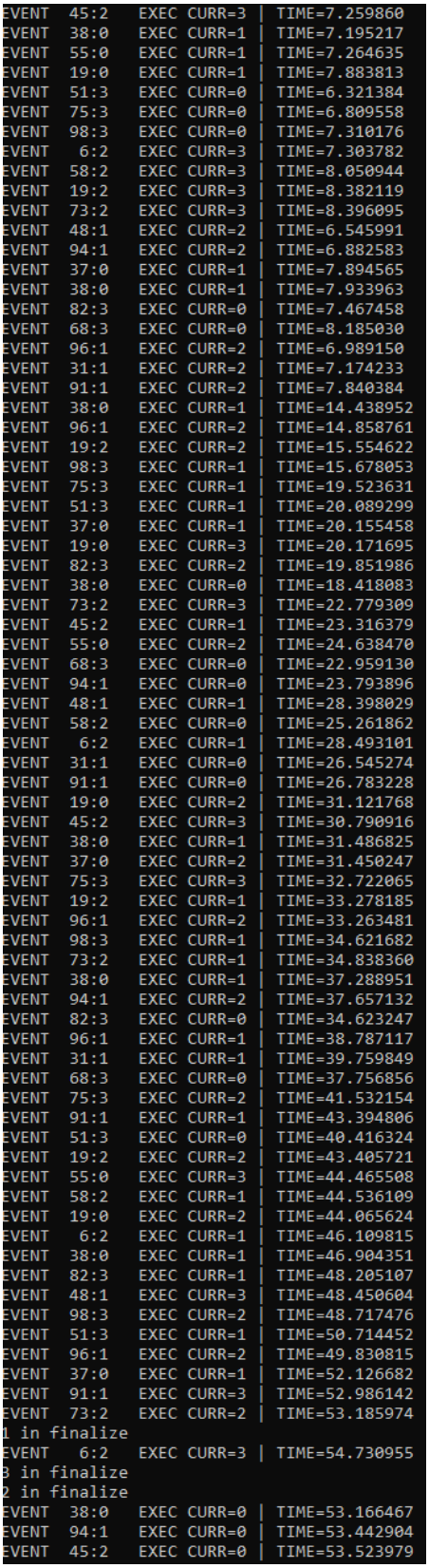

Figure 5. 2nd Test w/4-processors and 5-TestEA/process