Anthony Kim

Sara Khedr

Nick Fay

Anders Maraviglia

For our project, we will be simulating Lyme Disease using ROSS. ROSS will provide us with GVT instrumentation to keep track of global time throughout the simulation. It will also provide us with the event tracing capabilities. These capabilities will allow for each event to be kept track of so that computation can be executed at designated times within the simulation. The goal of this project is to build upon the existing work of Deelman and Szymanski, whose paper is linked below. In their model, they considered the spread of Lyme Disease through ticks and the movement of mice; we would like to also add deer to the model and create events tailored to the spread of Lyme Disease through deer.  In addition, we would like to see how this addition of deer will change our results in comparison to the results shared in the paper.

Initially, the model considers ticks at the early life stages. When a tick hatches, it is not infected because ticks cannot pass the disease onto their eggs. When the tick becomes a larva, it will find blood from the common white-footed mouse; if the mouse is infected, the larva will be infected as well. Next, the larva will drop off their host and become a nymph, which remains dormant during the winter. In the spring, the nymph will become active again and feed on mice. If the tick has previously been infected, it can now infect new hosts. The last stage is after its second blood meal, when a tick becomes an adult. In this stage, the tick feeds on larger animals, most commonly on white-tail deer. In our project, we will add the adult stage when modeling ticks; this includes defining an adult bite (along with the existing larval and nymphal). In addition, we will model the movement of deer and the percentage probability that the deer will be infected.

In terms of parallelizing the algorithm, we would like to split up the “space” between ranks. The simulation space will be a 2D array (wrapped in both directions), where each node represents the size of a mouse home range (400 m2). The space will be split into n sections when there are n processors used and each of the sections will use be assigned a logical process (LP); this closely models the paper cited below.

From this paper, it is clear that improving performance through parallelization of this problem is not straightforward; therefore, we do not find this project to be “embarrassingly parallel”. From the paper, it was found that the speedup grew for up to 10 processors and then began decreasing afterwards. Later, to improve performance, the simulation space was divided into more sections and less area was assigned to each LP. This again changed the speedup of the simulation with varying numbers of LPs and processors. We hope to run various tests and find the configuration that results in the the maximum speedup for our model.

Paper: https://www.cs.rpi.edu/~szymansk/papers/wsc96.pdf

Ok logging off now  Ok gnite

QUESTION: We good? Are we happy with this project, the scope, and the summary?

Nick’s Answer: i like it, i added some ROSS stuff then saw it was done

Yeah, thanks for looking into ROSS more. I got scared initially lol.

Anthony’s Answer:  Okay, I think this idea is really good in that 1) There is a paper to guide us  2) We get to use PDES, and event simulator, which is a good, cool thing to learn to use

3) Only thing I was worried about was if it wasn’t feasible, but I guess that won’t be a problem.

Should we ask Carothers tomorrow about whether the 2D lattice and “Strip Decomposition” refers to what we think it refers to? yes

Ok, I need to read more about this before I look dumb. I don’t know what an LP really is. In the project, they have 8 processors and 16 LPs. 1 processor to 2 LPs. Not sure if still 1 section to each LP.

QUESTION:  Why is Figure 3 a “Strip Decomposition”?  I googled it, and its not even a thing.

I think its just a  decomposition of thee space into strips lol they probably just used buzzwords

So this means that what you said earlier, they just split the 2D array into sections just like game of Life?

That’s what I think lol. I started writing that and wasn’t sure if they were the same, so I just used their terminology.

“The MPI message passing library (Gropp, Lusk, and Skjellum 1994) is used for communication between LPs. We use a strip decomposition in the **??dominant?? direction?** to divide the space (Figure 3). ***The number ofstrips is equal to the number of processors***. We assign one LP per strip. We chose this decomposition as the starting point for the system. As the research progresses, we will add other spatial decompositions”

For our Parallel Discrete Event Simulation part of the project,  our logical processes (LP) will consist of a state, event queue, and a clock.  The simulation will progress as events are removed from the queue.  An event causes a state change and the simulation clock advances.  MPI will be used to communicate between LPs.

“When a mouse starts moving in a certain direction, it will continue moving in the same direction until it settles at a nesting site”  ← does this mean it moves only from left to right in same row?  Or does it move up/down, diagnol?   Is this event simulator how mice starts from top left corner, and how it starts to infect from top left to bottom right?

I think it can just pick one of the 8 directions and it just keeps going in that direction. It might not start at the upper right corner, but it will define how the mouse moves at any spot in the grid.

For our group project, we would like to simulate the spread of Lyme Disease using ROSS.  ROSS is an acronym for Rensselaer's Optimistic Simulation System, and it is a parallel discrete-event simulator that executes on shared-memory multiprocessor systems.  ROSS is made for handling large-scale simulation models (from hundreds of thousands to millions of objects).

## Provide a 1 page summary on your group project. Describe your parallel algorithm approach (MPI, pthreads or both) and justify why's it not "embarrassingly" parallel.

Our group project is a simulation of a zombie apocalypse.. Each rank will have an undirected node graph.  This graph symbolizes a city.  At every tick iteration, nodes will be taken out of the graph and put into another graph.  These nodes would be put in randomly by creating edges between random nodes.  (MPI send/receive).  This will simulate people traveling between cities.  The two parameters the program will have is the number of nodes that are moved at each iteration, and the number of people a zombie can infect at each iteration.  Nodes (people) who are direct neighbors are susceptible of getting infected if one of their neighbors is a zombie.