Parasitic Species Containment Simulator (PSCS)

Mike Knowles
Marc Leef
Josh Schmidt
Evan Townsend

Executive Summary

As our civilization explores and expands our collective knowledge of the universe we are more and more likely to encounter biological agents or diseases that could penetrate traditional defenses, destroying our civilization. To help prepare for this reality we have designed a simulator to model different environment to simulate the outcome of projected parasitic attacks. Our initial tests of the simulator however revealed that despite vast amounts of computing resources, our simulation still could not keep up and project accurately. We needed parallelism to compute all the needed events while simulating multiple galaxies and updating a visual representation.

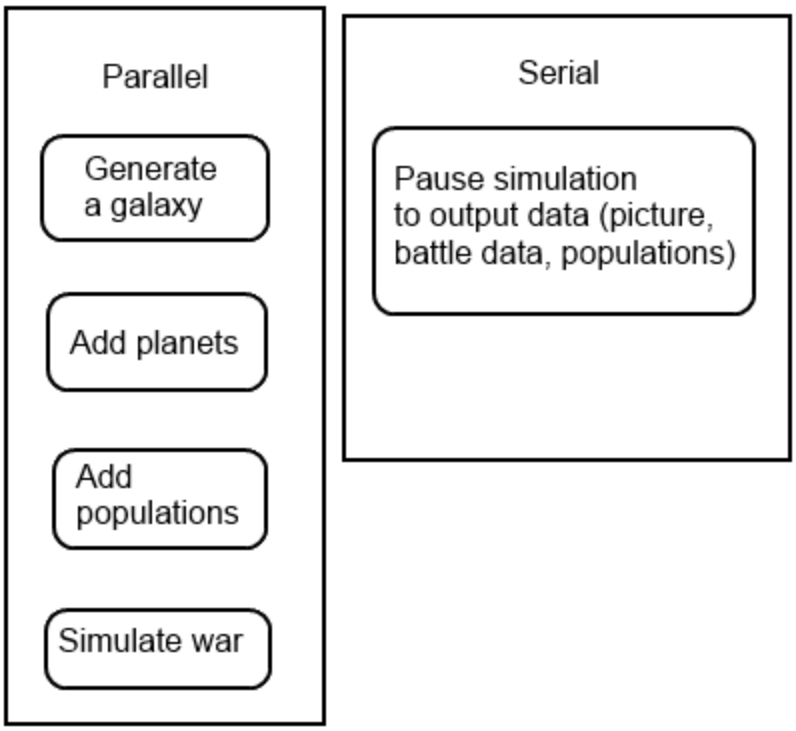
Project Description

With the increasing frequency of making first contact with previously unknown species, it is important to establish proper containment strategies should conflict ever arise. Over the course of thousands of years, our militaries have gotten very good at handling humanlike enemies, however biology is more creative. One particularly concerning threat is that of the intelligent parasite, a being that propagates itself in a manner similar to a pathogen, but can control its host and coordinate itself on the macroscopic level. The Parasitic Species Containment Simulator (PSCS) or 'Pices' is an attempt to model a conflict of this nature on a galactic scale, as well as compare a number of different strategies for its resolution

The simulation will consist of a galaxy divided into a number of sectors consisting of a number of planets at certain coordinates. Each clock tick will correspond to a number of hours or days, and travel between planets is dependant strictly on the distance between them. The objective the humans and the flood is the eradication of the other. As situations change within sectors, news of the events propagates outward and the surrounding sectors must react to protect themselves using a number of different strategies we will provide.

We will write a map generator, which will randomly construct a realistic model of a populated galaxy. We will then write a simulator which uses the constructed map, introduces the pathogen, and simulates the interaction between the two. The progress the invaders will be documented every few clock ticks by a function which compiles a graphical representation of the current state of the galaxy as well as population and attrition data. We plan to write a number of realistic offensive and defensive strategies so we can compare their relative effectiveness as well as long term costs. (For example: survival vs. planetary destruction) In the case where a strategy is able to defeat the enemy, we will adjust the virility of the pathogen to match.

Highlevel Architecture



Opportunities for Parallelism:

1 - Generating the Map:

The Universe will be divided into regions which contain a randomly generated set of planets with unique attributes. These subsections and sectors can be populated independently of one another by a parallel routine.

2 - Processing Updates to the Environment:

Similar to the universe generation, individual subsections and planets will be changing state every time tick. Values such as unit numbers or planetary resources will need to be updated, a task ideally suited to thread delegation. Threads may be dispatched in a hierarchical fashion to process these environmental updates and pass information up to parent threads.

3 - Generating a visualization of the map:

The rendering of the simulation will most likely take the form of a 3D rectangular prism with spherical representations of all the different planets. Different unit types (humans,flood, etc.) and their geographical density will be color coded so it becomes clear how individual skirmishes are going. Additionally, statistical information will be available about the state of the system. This rendering step can be performed in parallel.

4 - Testing Multiple Combat Strategies:

The ultimate goal of the project is to determine how to best distribute military forces, resources, and civilians to minimize loss of life and maximize destruction of the alien attackers. Tweaking starting parameters and the distribution of random variables, and the n subsequently executing these variants in parallel, will allow us to gain an in depth understanding of how best to handle the flood invaders.

Schedule

Week 5: Assign primary roles, select parallel language(s) and graphic rendering library
Week 6: Identify key events, design rough simulation environment, start map rendering
Week 7: Implement rough simulation
Week 8: Integrate rough simulator and map generator
Week 9: Fine tune simulator
Week 10: Finish up, work on presentation material