

Tasks Completed:

For this past week, some significant progress has been made. Gale is now up and running on the supercomputer with out errors. The big errors that were present initially dealt with output data. Disabling the checkPointEvery flag in the .xml file for each program got rid of the errors that occurred with .dat files. Also in the configuration file, the output path had to be changed to the absolute path, rather than the relative path so that the output data could be written correctly.

This past week, I also have been working on a specific example, tibet3D.xml, that causes Gale to crash with a segmentation fault. This example is the model that Ben will be using to base his 3D model on, so it is important to get it to work. Using the debugger, the segmentation fault occurs before or during the call to the GaleContext_AssignPointers(context, NULL); function in main.c. Figuring this out will take some inspection into the source code for Gale. If significant progress has not been made by the end of the next week, I plan on contacting CIG, which is the maintaining body for Gale.

Another, secondary task that I have been working on is getting Paraview to work on Linux. The precompiled binary that is provided on their website does not run properly on my machine, so steps have been taken to compile it locally. Cmake, a custom precompiler of sorts had to be installed, as well as Qt. Right now, cmake seems to be having issues locating the Qt executable, even if I manually set the path to point to it.

Data:

Below, in figures 1 through 4, is shown the speedup achieved when running one of the samples, yielding.xml, on the supercomputer.

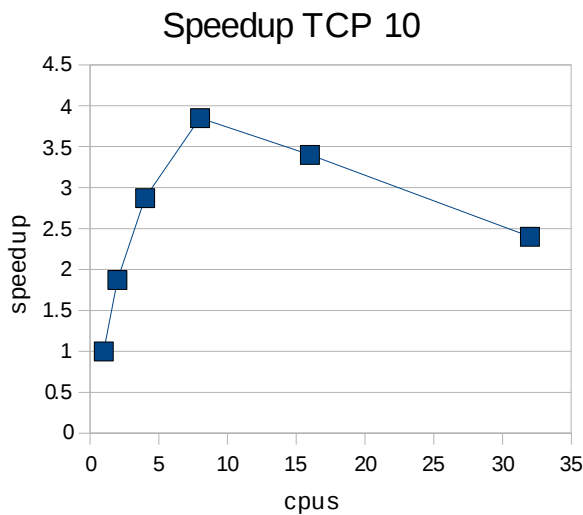


Figure 1. Ethernet, 10 time steps

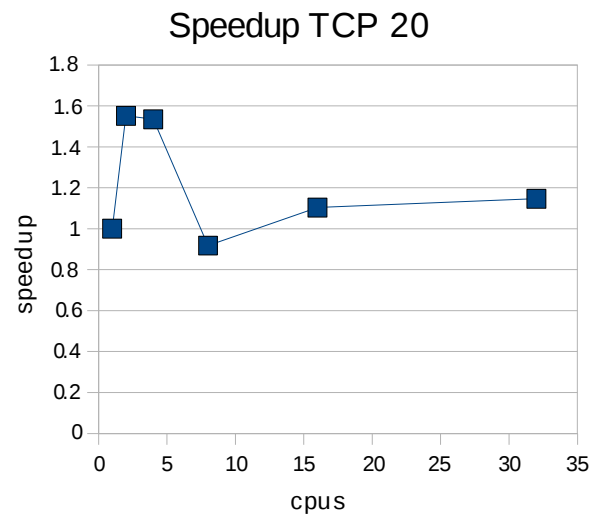


Figure 2. Ethernet, 20 time steps

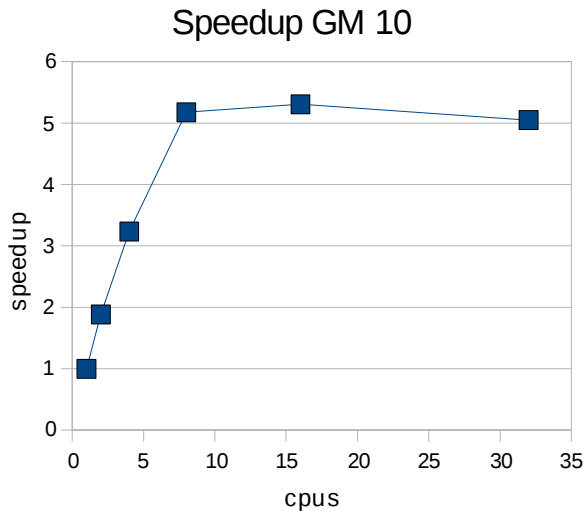


Figure 3. Myrinet, 10 time steps

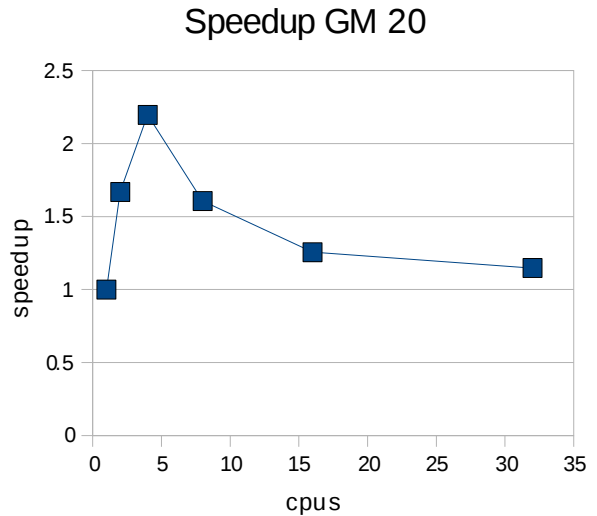


Figure 4. Myrinet, 20 time steps

From these figures, it is apparent from these figures that the optimum speedup occurs at 4 or 8 processors, and then either declines or stays relatively static for any more processors that are added. I have run jobs with most, if not all of the output disabled, and the trend seemed to be the same. In fact, when I disable the output, the run times for 8, 16, and 32 processors tends to increase a lot. This seems counter-intuitive if writing large files is the issues. Figuring out what is the cause of this and attempting to enhance the scalability of Gale is my next step.

Tasks for Next Week:

For next week, I would like to have Paraview up and running, as well as the tibet3D.xml program debugged and working correctly. Additionally, it is important to figure out how to scale Gale so that running it on a supercomputer is beneficial. Doing this will entail of a more thorough analysis of performance based on different numbers of processors, time steps, and alternating the amount of output that is produced by the program. Right now, my guess is that the additional communication between the processors when there gets to be about 16 to 32 of them is slowing the program down, but more complete data must be obtained before I can make this conclusion for sure.

In the long run, the goal is to get Gale operational at such a way that running it on 8 or more processors yields a significant speedup. Also, it needs to be able to run 3D models, which will be solved when the tibet3D.xml issue is taken care of hopefully. Viewing the data with Paraview must be done as well to check the validity of the data.