Objectives

- Create a framework for execution, analysis, and visualization of models.
- Exploit parallelism as a means of running multiple models simultaneously.
- Evaluate existing libraries and frameworks for parallelism across compute nodes.
- Present early findings and visualizations.

Introduction

VIPER presents a novel opportunity to exploit parallelism as a means of making scientific discoveries in previously unreachable areas. One such area is agent based modelling, specifically exploring parameter spaces in wealth distribution models.

By using relevant parallelization libraries we have been able to run multiple instances of Schelling’s model of segregation[4] with varying conditions and added mechanics. Outputs have then been visualized giving key insight into the model’s emergent properties and the relationships between their parameters. Figure 1 shows an example of a single model run, with clusters appearing as shaded regions. All presented figures are derived from experiments run on a 500x500 grid of agents over 1x10^6 time steps which have been normalized.

Early Methodologies

As preparation for deployment to VIPER, multiple test versions of bespoke agent based modelling frameworks have been developed initially for use on local machines. These vary greatly in terms of their overall architectures and the languages/libraries they incorporate. Earliest work included a wrapper style program (C++) where each model would be called as a standalone program by another ‘launcher’ script. Tailored programs were then consolidated into a single unified framework with parameters set in the code as opposed to through arguments.

With successfully parallelized models running efforts were made to make the process of experimentation more concise. This meant streamlining our workflow and exploring extra software and techniques to reduce the time taken from inception of a new model idea to successful execution and analysis. Through implementation of cloud based storage, collaborative code editing, and a number of other measures we have been able to condense the preliminary exploration process down to a matter of minutes as opposed to days.

VIPER Utilisation

Early VIPER usage was limited to single and multiple processors on a single node using the OpenMP library for inter-process coordination. As exploration requirements increased, OpenMPI implementations were explored allowing increased usage as high as 100 processors over multiple nodes. This work presents potential to again increase usage by a similar factor, facilitating more rigorous statistical analysis as subsequent outputs are collated.

Usage over the evolution of this project translates to approximately 50 core hours across multiple users with each run taking approximately 10 minutes of real time to execute. Figures 1 through 4 represent the results of over 40 job submissions, each more computationally intensive than the last.

Visualization & Analysis

Model outputs are stored as CSV files, saved directly to the VIPER file system. These files are then transferred to a local machine for visualization and analysis using several Python scripts. As exploration efforts increase so will the need for robust and efficient visualization methods, making transfer of the data visualization computation to the VIPER nodes important future milestone for this project.

Figure 2 presents an initial method of analysing the data, such ‘slices’ of the surface plot style data in figures 2 and 3 offer the ability to extract measurements which can then be plotted to give a much higher level view of the findings of our experiments. These include typical statistics measures such as skew and kurtosis and can be built upon over time.

Conclusion

Through utilising VIPER we have found a rich and interesting area for further work, with benefits beyond this project alone as the code base can be repossessed for other applications. The developed framework could also be shared between researchers both in the same groups and across departments within the university, promoting inter-department collaboration and increased utilization of the high performance facilities.

Future work will include implementing more sophisticated analytical and visualization capabilities with an emphasis on increasing utilization of VIPER as models increase in computational demand.

References