Assignment #4: Data Collection and Preparation

Caitlin Ross & Noah Wolfe
02/18/2016

1 Data Source (ROSS)

In this assignment, we have decided to generate/collect data from our ROSS research projects. ROSS is a massively parallel discrete-event simulator that can process billions of events per second [2], [1]. ROSS models are made up of a collection of logical processes (LPs). Each LP models a distinct component of the system. LPs interact with one another through events in the form of time-stamped messages. An MPI task is abstracted as a processor element (PE) in ROSS. Each PE owns a number of LPs and schedules events in time-stamp order for all LPs assigned to it. Events that are destined for a logical process on another PE (i.e. remote events) are sent as MPI messages.

The underlying ROSS simulator is itself, a very complex piece of software with a large set of parameters. When changed, these parameters can trigger large changes in metrics such as number of reverse computations, remote events, event efficiency, etc. The only data collection is averaged over all processes and collected at the very end of the simulation.

2 Research Questions

ROSS is designed to be a very fast and efficient simulator. In order to get the best possible speed-up, developers need to know what is going on behind the curtains. We need to know not only if a simulation is experiencing hotspots, but also be able to detect the cause of those hotspots. This requires collection of data throughout the simulation. Two research questions that can be solved by analyzing this data are:

- Is my ROSS model keeping a uniform distribution of computation among MPI processes.
- Which ROSS scheduler parameters are the most influential to the speedup of

With some data manipulation, we could find pinpoint key

3 Hypotheses

4 Data Format, Extraction & Manipulation

Using the simple csv format, the raw data will be structured so that the columns represent the various ROSS parameters and metrics and the rows represent those same parameters collected at a later instance of simulation time. The default sampling frequency is $0.01 * simulation\_end\_time$ but can be adjusted with a commandline flag "–report-interval=n" where $n \in 0.999, ..., 1/simulation\_end\_time$. Each MPI process creates it's own data file and appends the MPI rank number to the end of the filename. Filenames can be adjusted with the commandline flag "–stats-filename=string".

5 Sample Visualizations

6 Visualization Tool Review

References

[1] D. W. Bauer Jr., C. D. Carothers, and A. Holder. Scalable time warp on blue gene supercomputers. In *Proceedings of the 2009 ACM/IEEE/SCS 23rd Workshop on Principles of Advanced and Distributed Simulation*, PADS '09, pages 35–44, Washington, DC, USA, 2009. IEEE Computer Society.

[2] A. O. Holder and C. D. Carothers. Analysis of time warp on a 32,768 processor ibm blue gene/l supercomputer. In *in 2008 Proceedings European Modeling and Simulation Symposium (EMSS*, 2008.