Parallel Data Streaming with GStreamer and MPI

Garrett Roberts: Brigham Young University—Idaho
Jian Tao: Louisiana State University

Abstract
Data streaming is the movement of data from one place to another. To accomplish more in the same time, we use parallel processing, which uses multiple processors simultaneously. Depending on the methods used, this can improve both latency, the delay between input and results, and bandwidth, the amount of data that can be processed simultaneously. This project uses a data streaming framework, GStreamer, and a parallel processing framework, MPI, to achieve parallel data streaming and processing.

The Parallel Streaming Plugin
The result of the project was a collection of new elements for GStreamer, or a plugin, called “GstMPI.” These elements use MPI functions to communicate with pipelines on other processes. Thus, each process can run its own pipeline independently, using MPI functions to coordinate as necessary, for example, sending data to analyze or returning the results of number crunching.

GStreamer is a multimedia streaming framework. It links individual elements together to create pipelines.

The Message Passing Interface (MPI) is a standard for a distributed memory parallel computation framework. It is designed for systems whose CPUs do not share the same memory, hence “distributed memory.” It provides an API filled with functions to pass data among many processes over a network.

Time Series Analysis
A time series is any set of data points collected on a scale of time. These types of datasets have applications in signal processing, coastal modeling, financial data analysis, gravitational wave analysis, etc.

MPI

Network

CPU

Message

CPU

Memory

CPU

Memory

CPU

Memory

Future Work
While the framework developed in this project is in need of further development and extension, it can be used directly in time series analysis applications. Specifically, it may prove to be a useful tool in future coastal modeling projects.

References
http://gstreamer.freedesktop.org/ (Images)

Acknowledgements
This material is based upon work supported by the National Science Foundation under award OCI-1005165 with additional support from the Center for Computation & Technology at Louisiana State University.