High Bandwidth Data Collection and Processing Using OpenMPI on a LINUX Cluster

David L MacPherson and Greg Kliss
SRC
6225 Running Ridge Road, North Syracuse, NY 13212
dmacpherson@syrres.com, gkliss@syrres.com

Introduction

Processing the data streams from a high speed data collector in real time requires substantial processing resources. This paper provides an experimental software architecture that runs on a multi-core high performance computing cluster using OpenMPI. The objective of this paper is to describe the data collection and processing experiments. This paper provides measurements the processing capability for certain software architectural configurations. A Linux high performance computing cluster with 80 cores and 320 GB of main memory was used for these experiments. The overall goal is real-time processing of the data collector's data stream supporting the design of a deployed on a mobile collection platform.

Objectives

Investigate ultra high bandwidth data collection and processing including:

- Benchmark sample processing time for various processing topologies.
- Prototype software architectures most suitable for the high input data rate.
- Design a data collector and processing system that monitors for short burst transmissions in a given frequency band.

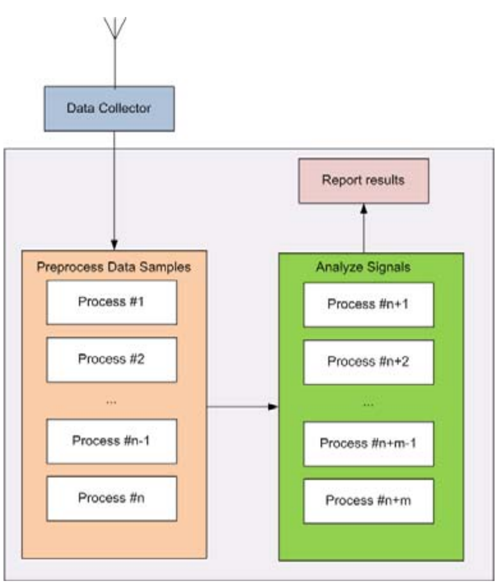

Figure 1 - Software Architecture

Software Architecture

The system architecture includes a data collection device and a processing cluster. The software architecture of the cluster program uses OpenMPI to achieve multi-processing on 80 cores. The software architecture is shown below in Figure 1. The data collector feeds the preprocessing pipeline that consists of n separate processors. Data is passed from preprocessor to m signal analysis processors. The data transfer buffer size, n and m were modified in software to achieve the maximum processing bandwidth.

The of LINUX cluster

The processing platform is a cluster of ten Dell model 1950 servers with eight Intel® Xeon processors each. This cluster has 320 GB of memory and 80 cores. A diagram of the cluster is shown in Figure 2.

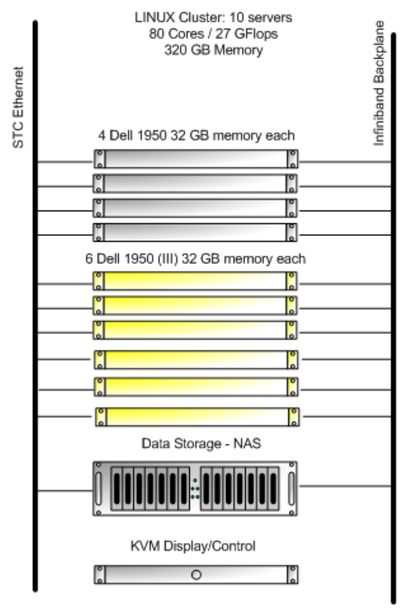

Figure 2 - The SRC High Performance Computer Cluster

SRC RF Data Collector

The SRC ultra wide band data collection system is a custom-made system designed for field data collection. The data collector device is designed to operate in the 500 MHz to 18 GHz frequency range with a 300 MHz bandwidth. Additional broadband down converters on the front end could extend this frequency band. A diagram of the data collector is shown in Figure 3.

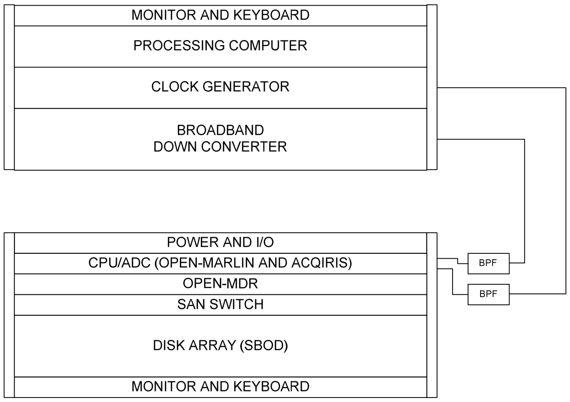

Figure 3 - Ultra Wide Band Data Collection System

The data collector stores 720 million samples per second. The data is stored as one byte for each sample. This paper will describe various processing strategies for distribution and parallel processing of the incoming signal. Some of the challenges associated with moving large blocks of data through the processing chain are examined. A strategy for efficient data processing using 80 cores is provided.

Results

A plot of the signal at 168 MHz is shown in Figure 4. The data collection and processing systems were optimized to allow monitoring of certain portions of the frequency spectrum. Alerts could be added to let operators know when a signal exceeds a certain threshold level. Individual or groups of processor could be dynamically re-assigned to monitor important portions of the frequency spectrum when required.

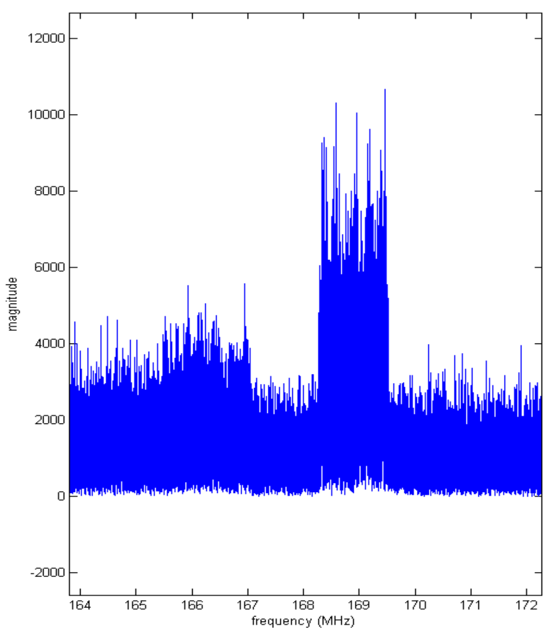

Figure 4 - Interesting Signal at 169 MHz

The inter-process data transfer rates for various buffer sizes is shown in Figure 5. Data transfer rates were higher when larger buffers were used. Real time processing of the data stream requires the system to handle 720 million samples per second, however, with 80 processors each handling at least nine million samples per second, processing data in real time becomes possible. Experimental results show that when the data transfer buffer size is 2 MB or larger, the data processing rate supports real-time data collection. Higher data processing rates might be achieved using a tree data distribution scheme. [1] Signal processing algorithms were constructed from the GNU open source code library. [2]

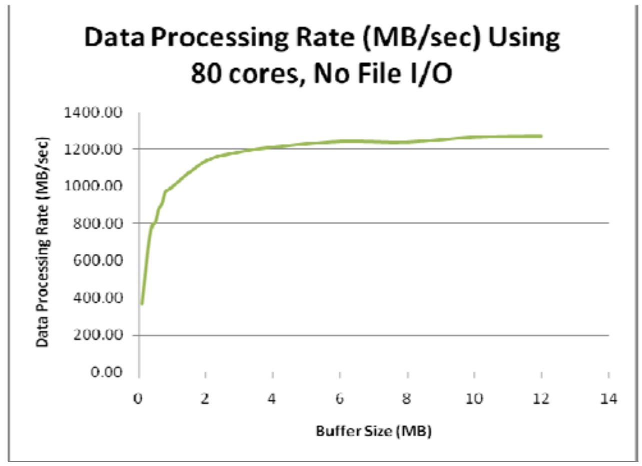

Figure 5 – Data Processing Rate Versus Buffer Size

File I/O was a substantial bottleneck when the processed data was saved. In order to match the data collection rate, the sample buffers were retained in main memory.

In spectrum plot of the sample used for this study is provided in Figure 6. The local FM radio stations in Syracuse, NY standout near 100 MHz.

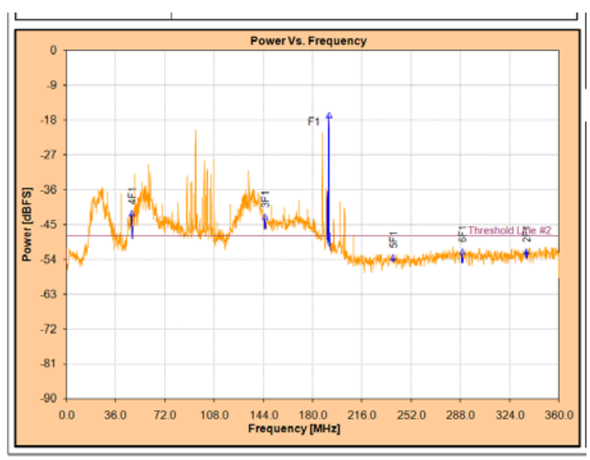

Figure 6 - Power versus Frequency Plot

References

[1] P. Pacheco, *Parallel Programming with MPI*, Morgan-Kaufman, 1997.

[2] M. Galassi, et. al., "GNU Scientific Library Reference Manual," 3rd edition, Network Theory Ltd, 2009.