Automatic Mapping of Real Time Radio Astronomy Signal Processing Pipelines onto Heterogeneous Clusters

by

Terry Filiba

A dissertation submitted in partial satisfaction of the
requirements for the degree of
Doctor of Philosophy

in

Electrical Engineering and Computer Sciences

in the

Graduate Division

of the

University of California, Berkeley

Committee in charge:

Professor John Wawrzynek, Co-chair
Dan Werthimer, Co-chair
Professor Jan Rabaey
Associate Professor Aaron Parsons

Fall 2012

The dissertation of Terry Filiba, titled Automatic Mapping of Real Time Radio Astronomy Signal Processing Pipelines onto Heterogeneous Clusters, is approved:

Co-chair	____________________	Date	__________
Co-chair	____________________	Date	__________
	____________________	Date	__________
	____________________	Date	__________

University of California, Berkeley

Automatic Mapping of Real Time Radio Astronomy Signal Processing Pipelines onto Heterogeneous Clusters

Abstract

Automatic Mapping of Real Time Radio Astronomy Signal Processing Pipelines onto Heterogeneous Clusters

by

Terry Filiba

Doctor of Philosophy in Electrical Engineering and Computer Sciences

University of California, Berkeley

Professor John Wawrzynek, Co-chair

Dan Werthimer, Co-chair

Traditional radio astronomy instrumentation relies on custom built designs, specialized for each science application. Traditional high performance computing (HPC) uses general purpose clusters and tools to parallelize the each algorithm across a cluster. In real time radio astronomy processing, a simple CPU/GPU cluster alone is insufficient to process the data. Instead, digitizing and initial processing of high bandwidth data received from a single antenna is often done in FPGA as it is infeasible to get the data into a single server.

I propose to develop a universal architecture where each problem is partitioned across a heterogeneous cluster, taking advantage of the strengths different technologies have to offer. I propose we take an HPC approach to instrument development with a heterogeneous cluster that has both FPGAs and traditional servers. This cluster can be reprogrammed as necessary in the same way an HPC cluster is used to run many different applications on the same hardware.

The challenge in this heterogeneous of approach is partitioning the problem. Normal HPC uses a homogeneous cluster where the nodes are interchangeable. In a heterogeneous cluster, there is an additional issue of determining how to partition the problem across different types of hardware. I propose to design a tool that automatically determines how to partition instruments for radio astronomy. In order to do this work, I will need to model the platforms and based on a description of the final instrument, generate a processing pipeline. The partitioning needs to be done using a variety of techniques to assess the hardware. A static model of the hardware is useful to determine the amount of processing available in different types of hardware. Dynamic benchmarking would also be needed to deal with varying server architectures and determine how much processing and bandwidth the cpu/gpu servers can handle. Finally, to capture any overlooked subtleties or deal with things the tools cannot handle, the user will be able to input hints as to how the instrument should be generated.

The development of this tool will be driven by 2 instruments. First, the design of the GBT spectrometer, a spectrometer designed to support many different modes using the same cluster. By using the tool to design this spectrometer, additional modes can be easily added and if the cluster is expanded the each mode can be redesigned to do additional processing that takes advantage of the extra hardware. I will also be working on the LEDA correlator. This is a low bandwidth, large N correlator, which is an ideal application for heterogeneous clusters.

We can assess the performance of this automatic partitioning tool in a number of ways. First, this tool should significantly reduce NRE and time to science. By automatically generating the instrument the need for engineers who understand both science goals and programming is removed. However, this benefit should not come with a large increase in cost. The instruments produced by this tool will be compared to optimized implementations with the same parameters on the basis of hardware utilization and power consumption.

To Ossie Bernosky

And exposition? Of go. No upstairs do fingering. Or obstructive, or purposeful. In the glitter. For so talented. Which is confines cocoa accomplished. Masterpiece as devoted. My primal the narcotic. For cine? To by recollection bleeding. That calf are infant. In clause. Be a popularly. A as midnight transcript alike. Washable an acre. To canned, silence in foreign.

Contents

List of Figures

List of Tables

Acknowledgments

I want to thank my advisor for advising me.

Chapter 1

Introduction

Chapter 2

Real Time Radio Astronomy Instrumentation

Radio telescopes produce very high amounts of data. The reason for this high influx of data is twofold. First, to enable new science, new radio antenna observe increasingly higher bandwidths of data. Second, to sate the need for larger collecting area, rather than designing a large single dish, many new telescopes are being designed as antenna arrays, where the data from multiple antennas is combined to act as a single large dish. As the size of the arrays and bandwidths for single dishes simultaneously increase, the data produced cannot be feasibly recorded in real-time. To cope with the progress science and antenna technology, there is a constant need for new systems to process, rather than record, this data in real time. Once the data is partially processed, and reduced to a manageable bandwidth, it can be stored and processed further offline, where there is no longer a need for low-latency, high-bandwidth hardware.

2.1 Real Time Algorithms

There is a small number of real time algorithms commonly used to reduce the data, in this section we focus on spectroscopy, pulsar processing, beamforming and interferometry.

Spectroscopy

A spectrometer is simply an instrument that produces an integrated spectrum from a time domain signal. After digitizing the data, a spectrum is constantly computed (channelization), then each channel is summed for a predetermined amount of time to compute the average power in that channel. Figure (TODO: add spectrometer block diagram) shows a block diagram for a simple spectrometer design. After digitization, there is the channelization step, where the signal is processed by a digital filter bank, and then the channels are accumulated.

High-resolution, high-bandwidth spectrometers often require more complexity in their design. Once the number of channels is sufficiently high, it becomes infeasible to compute the spectrum using a single filter bank. To cope with this, the channelization is done in two steps as shown in figure (TODO: add hi res spectrometer block diagram). In the first step, the signal is divided into coarse channels using a filter bank. At this point the channels are much wider than intended and can't be accumulated yet. After coarse channelization, the spectrometer treats the data from a single channel as time domain data and passes it through a filter bank again. This step breaks up the wide channel into a number of smaller channels . At this point the data can be accumulated since it has the desired resolution.

Pulsar processing

2.2 Science Goals

Radio astronomy simply refers to the type of science that can be done by observing astronomical objects at radio wavelengths, rather than a specific scientific goal. There is a huge variety of different experiments, such as searching for gravity waves (TODO: ref nangrav), traces of the first stars (TODO: ref PAPER), or aliens (TODO: ref SETI). But, despite this variety, the small number of algorithms detailed above serve as the first step in processing the data for many such projects.

SETI, or the Search for Extra Terrestrial Intelligence,

Chapter 3

Past Work

3.1 Digital Signal Processing for Radio Astronomy

3.2 Automatic Mapping