SOFT354 - Parallel Computing and Distributed Systems

A comparison of the Discrete Fourier Transform algorithm implemented in CUDA and MPI.

Ben Lancaster
January 12, 2018

Abstract

My placement as a Firmware Engineer at Spirent, a world leader in GNSS simulators. Coming from a Computer Science I was up to speed on the programming aspect of firmware, however was lacking in experience of electronic lab equipment. Constant exposure to this new area of technology and equipment has greatly improved my knowledge in GNSS and embedded programming and has influenced my future career choices. I was chiefly responsible for writing a new programmable timer, implementing fan-control strategies and power calibration schemes, writing a driver to read in peripheral devices, provisioning a Linux hypervisor, and designing and implement a Linux USB driver.

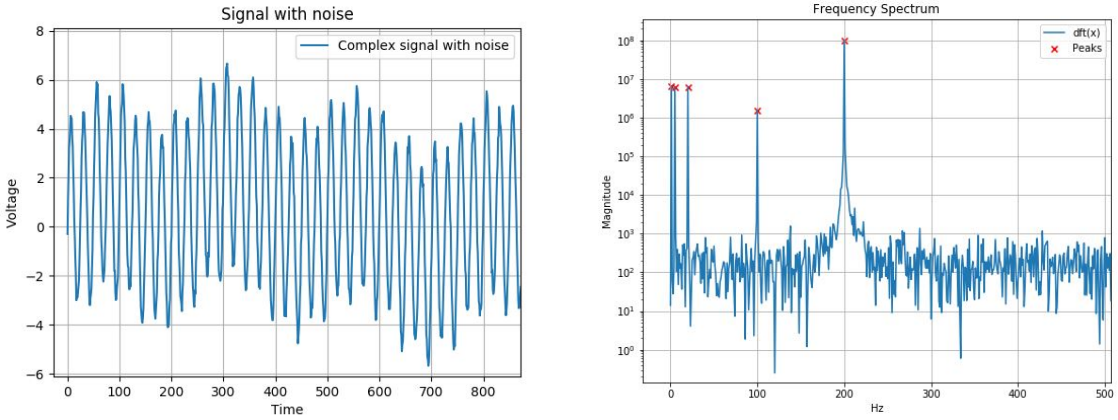
Table of Contents

1 Introduction

This report discusses the implementation and performance of the Discrete Fourier Transform (DFT) algorithm in CUDA and MPI.

DFT is largely used in digital signal and image processing applications.

In this report, I will be implementing the DFT to convert signals in the time domain to the frequency domain.



(a) Time domain **(b)** Frequency domain

Figure 1: Left: A compound signal made up of 1, 5, 20, 100, and 200 Hz sine waves in the time domain. **Right:** Frequency domain representation showing high amplitude peaks for the 1, 5, 20, 100, and 200 Hz waveforms.

2 Implementation

2.1 CUDA

2.2 MPI

With DFT, each sample must have operate on all other samples. In my implementation, I must first copy the complete sample array to each node.

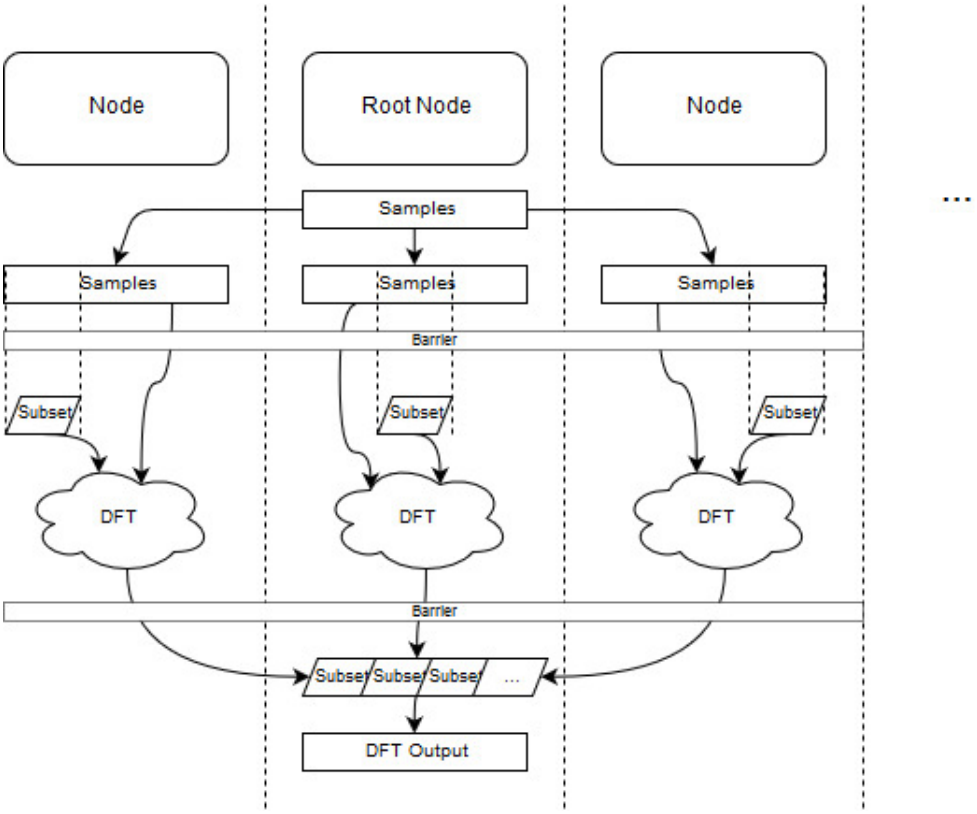


Figure 2: Control flow diagram for the MPI DFT algorithm.

3 Evaluation

3.1 Measuring Performance