

Computing the Proper Orthogonal Decomposition in Parallel : Study and Implementation

Govind Gopakumar

We describe the progress in an ongoing effort to study the proper orthogonal decomposition. The aim of the project is to study the speedups we can achieve in computing the POD in parallel, as opposed to traditional methods. This project also includes the development of a POD library, implemented on GPGPUs, or General Purpose Graphic Processing Units. This is a mid term report of the project, submitted in partial fulfillment of the requirements of the B Tech Project, Department of Aerospace Engineering, IIT Kanpur.

I. Introduction

As part of the requirements of the course AE471A, this project is being undertaken in two parts. Part A consists of the study of the technique known as the Proper Orthogonal Decomposition, and the benefits and pitfalls associated with implementing it on a parallel architecture. Concurrently, we develop a library that aims to implement the Proper Orthogonal Decomposition in parallel, using the CUDA environment provided by Nvidia. In this mid term report, we outline the work that has been carried out so far, and the proposed plan for the project execution.

II. Background

The Proper Orthogonal Decomposition is used widely in different fields, for reducing the dimensionality of the data being handled.

III. Motivation

There is a vast range of application for the technique known as POD. In Signal Processing, Data Mining, Statistical Learning, it forms a core of a set of methods that help in dealing with large amounts of data. Specifically in computational fluid dynamics, we can deal with large scale descriptions of flow by proper decompositional modes, which significantly reduce our computational effort, while providing us a nearly accurate solution. If there exists avenues to speed up the calculation of the POD, it would help boost the speed of current techniques and help in larger adoption of this technique.

Computing the POD of a dataset is relatively fast when we use low level languages like Fortran, and with usage of systems like MPI, we can effectively compute the POD for large datasets in a trivial amount of time. This is possible largely due to the presence of high performance multicore computers, which the majority of students do not have access to. The aim of the library developed as part of this project is to create an easy to use and fast library for general usage, with possible extension into newer languages like Python. This would ensure that newcomers to the field can quickly utilise this library off the shelf, instead of using Fortran code which is comparatively tough to understand.

One of the main objectives of this project is to check for possible speedups over the existing library. Initial studies show that trivial matrix based calculations can be optimized to provide up to a 20x speedup over traditional CPU based computation, and it would be one of the aims of this project to try and obtain such massive improvements in the existing framework.

IV. Literature review

We give a brief overview of the current work in this field, as well as the relevant work which has been used to aid this project. A comprehensive overview shall be present in the final report.

IV.A. POD

IV.B. POD in Aerospace Engineering

IV.C. Parallelization and CUDA

V. Theoretical overview

V.A. CUDA

CUDA, or Compute Unified Device Architecture is a parallel computing platform created by NVIDIA corp. It allows using general GPUs present in mid-high end computers for computation. The advantage of using this over traditional CPU based computation is two fold:

- There are a large number of compute cores in a typical GPU as compared to a typical CPU. A low end GPU contains upwards of few hundred cores, as compared to a high end CPU which may contain less than 20 cores.
- These compute cores are highly interlinked and optimized for parallel execution. They are most effective when used for trivial computation that can be massively parallelized.

CUDA has come to prominence of late because of the support Nvidia has released, of developing general computing applications on their hardware. Usage of CUDA for scientific computing is slowly gaining favour with researchers, as it provides a cheaper and faster alternative to bigger larger CPU based systems.

V.B. POD

VI. Work outline

We plan to carry out the project in three phases:

Initialization: This would involve a time period up to the mid semester examination. The work carried out in this period would be primarily the initial configurations, and theoretical study. An initial literature review would be carried out, to identify the relevant work, as well as the state of the art. Also, the main computing system would be set up for development of the library. Existing code would be tracked and stored for further studies.

Internalization: This would be involve the time period from the mid semester examination till October end, roughly a period of 5 weeks. A concrete study of the theory would be finished by this time, and the program layout finalized. The most appropriate relevant research work would be identified as the basis for the project, and existing code would be analyzed thouroughly, for improvements and application specific optimization.

Realization: This would be carried out in the month of November, culminating in the final project report, as well as release of relevant code. The report would carry an overview of the results obtained, the relevance of the results, all code that was developed, as well as relevant documentation for the code.

References