Self-accelerating Processing Workflows (GPU vs CPU with insights to optimize)

Supervisors:

- Janaka Alawatugoda (LSEG)
- Dr Adeesha Wijayasiri

Group Members:

- B.Abinayan
- J.Jeyakeethan
- T.Nirojan

Introduction

- GPGPU has great trend in latest general purpose computing.
- It provides a gain of reduction over execution time for some computations over huge dataset.
- GPU performs well for SIMD but CPU is necessary for serial part of the code (MIMD).
- The right choices of Platform may save a long time in accumulation.
- Bandwidth is a bottleneck for a GPU to achieve the gain.
- Arithmetic intensity is a good measure to determine its suitability.

Research Problem

- Many computations can be done either on CPU and GPU platforms.
- The computation time of the problem may increase by a factor if a problem is executed on a wrong platform.
- A tool to predict the right platform in runtime that do a computation in low execution time.
- Different computations types shows different performance boundaries in between the CPU and GPU.

Research Motivation

- The right choice may reduce overall execution time by a certain factor.
- The selection of platform cannot be done dynamically based on computation type and there is no option to execute different codes on the CPU or GPU upon the selection.
- Few related researches have been conducted previously but none of them provide an appropriate solution.
- A library that evaluate a computation to determine the suitability would help to take a optimum choice in runtime.

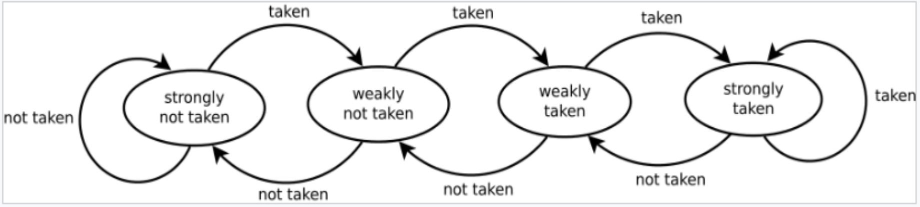
Research Objectives

- A library of functions that predict the right platform for a defined computation.
- The functions to define characteristic of the computation.
- An overall gain of execution time.
- The gain of execution time at cost of at least resources.
- The objective is not to utilize resources efficiently.

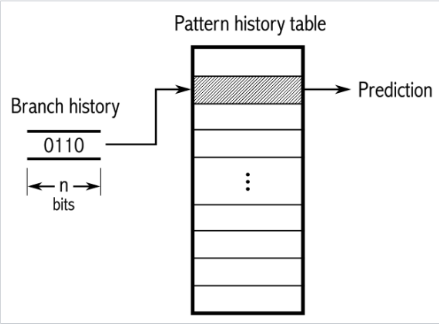
Branch Predictor (2-bit) Similar Concept

- A digital circuit to predict the way, a branch would take in advance.
- Modern branch predictors has an accuracy of 95%.
- It provides speedup gain more than 1.5 times.
- Classified into four types primarily.

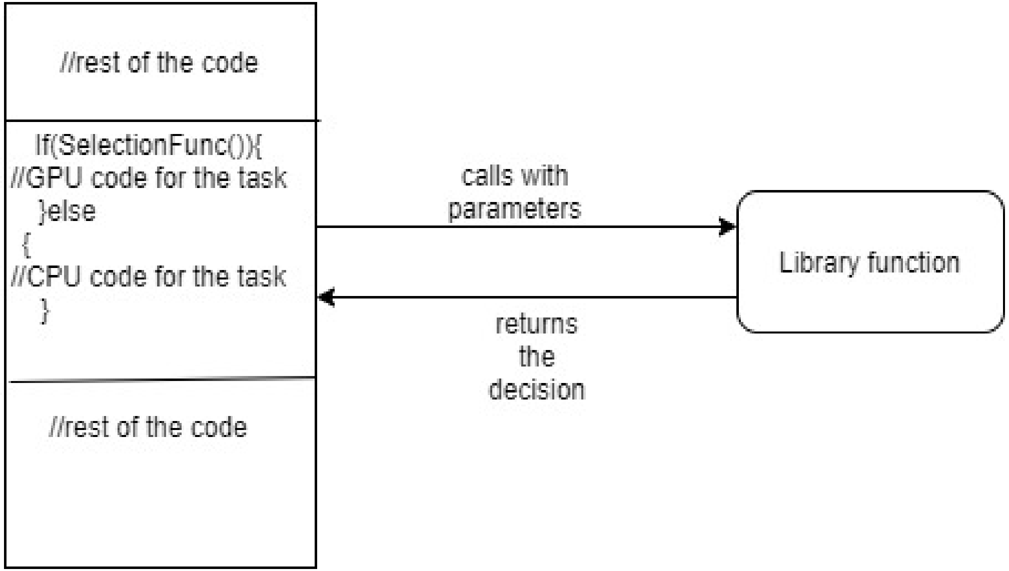
*Static *Dynamic *Random *H



2-bit state machine



Model of Solution



```
//rest of the code
```

```
If(SelectionFunc()){
//GPU code for the task
   }else
 {
//CPU code for the task
   }
```

```
//rest of the code
```

calls with parameters

returns the decision

Library function

Previous Art

- Though there are researches to optimize the performance by scheduling properly, it does not have more related previous works.
- Previous works mostly support as evidence to identifying the attributes that affect the performance of CPU and GPU.

Previous Arts (continued)

- The results of the experiments showed that the amount of data and the number of available parallel instances have a decisive influence on which platform is more efficient. [1]

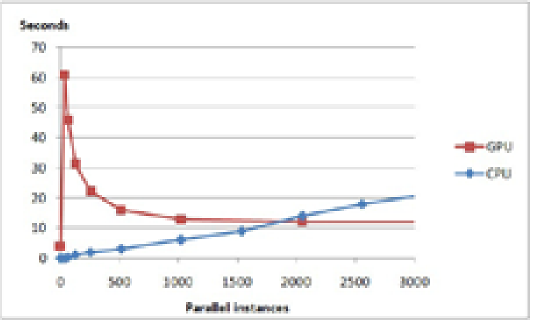


Figure9.Computational cost in relation to the number of parallel instances when a small amount of data is handled.

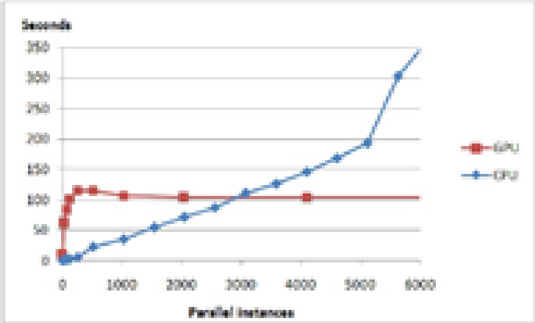


Figure 7. Zoomed-in view of graph presented in Figure 6.

Previous Arts (continued)

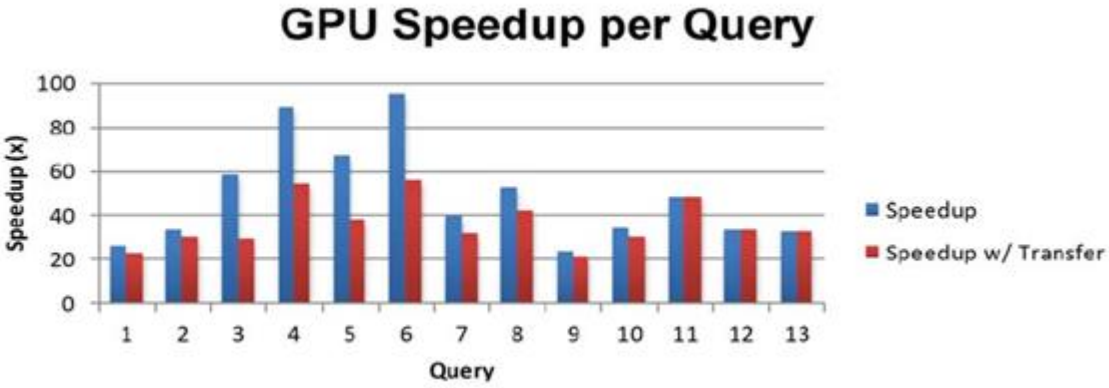
- Optimizations for CPU that contributed to performance improvements are: multithreading, cache blocking, and reorganization of memory accesses for SIMDification.
- Optimizations for GPU that contributed to performance improvements are: minimizing global synchronization and using local shared buffers are the two key techniques to improve performance

Previous Arts (continued)

- Rootbeer is a project that allows developers to simply write code in Java and the (de)serialization, kernel code generation and kernel launch is done automatically
- Rootbeer supports features of the Java Programming Language, that are (1) single and multi-dimensional arrays of primitive and reference types, (2) composite objects, (3) instance and static fields, (4) dynamic memory allocation, (5) inner classes, (6) synchronized methods and monitors, (7) strings and (8) exceptions that are thrown or caught on the GPU. [10]

Previous Arts (continued)

- ❖ Despite this variation, the minimum speedup for the 13 queries considered was 20X



Methodology

- Consist of 5 step process

  1. Extracting features
  2. Prioritizing features
  3. Establishing relationship
  4. Formation of methods
  5. Implementation techniques

Extracting features

- Identifying the attributes that affects the performance of CPU and GPU.
- Such attributes are planned to be taken from,

  1. Previous works
  2. Research papers
  3. White papers of CPU and GPU

  E.g : Array size, Array dimensions

Prioritizing Features

- Ordering the features based on their impact level.
- Helps in trade-off situations.
- Filtering the hardware level features from software level features.
- The filtering helps for the generalization of the library .

Establishing Relationships

- Measuring the impact level of attributes.
- The impact level will be measured for each attributes both individually and groups.
- Measuring process is planned to be conducted through statistical analysis.
- Tool: Matlab graphs

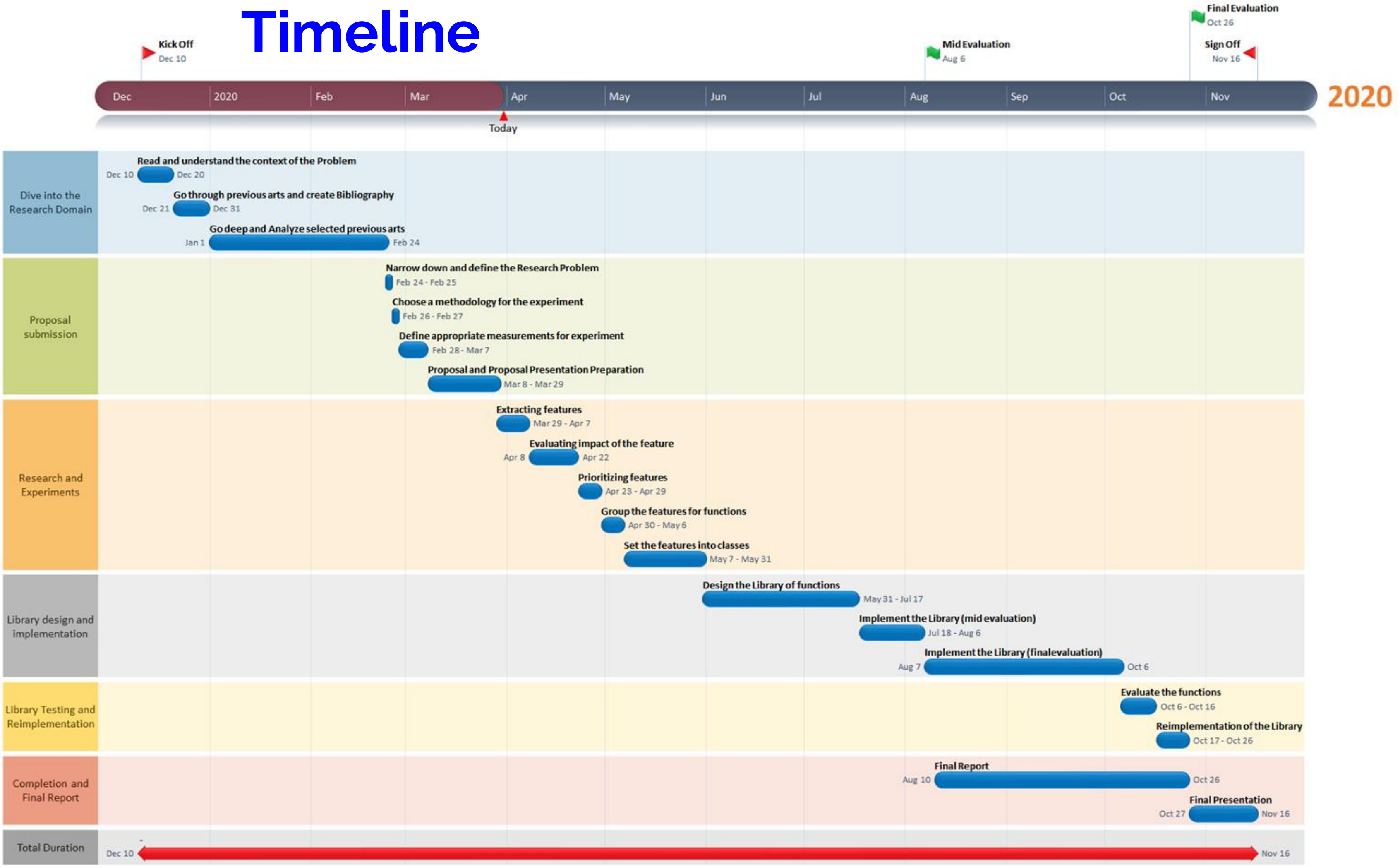
Formation of methods

- Based on the impact level measurements, the attributes with similar flow of pattern and context will be grouped.
- Selection methods/functions will be formed for those features.
- I.e. there will be parameters for a particular function.

Implementation Techniques

- Logical implementation of the methods will be designed in this stage.
- Appropriate algorithms will be formed for the selection of the platform.
- Algorithms will make decision based on the benchmarks obtained for each attribute through experiment.

Timeline



Dive into the Research Domain

- Read and understand the context of the Problem: Dec 10 – Dec 20
- Go through previous arts and create Bibliography: Dec 21 – Dec 31
- Go deep and Analyze selected previous arts: Jan 1 – Feb 24

Proposal submission

- Narrow down and define the Research Problem: Feb 24 - Feb 25
- Choose a methodology for the experiment: Feb 26 - Feb 27
- Define appropriate measurements for experiment: Feb 28 - Mar 7
- Proposal and Proposal Presentation Preparation: Mar 8 - Mar 29

Research and Experiments

- Extracting features: Mar 29 - Apr 7
- Evaluating impact of the feature: Apr 8 – Apr 22
- Prioritizing features: Apr 23 - Apr 29
- Group the features for functions: Apr 30 - May 6
- Set the features into classes: May 7 - May 31

Library design and implementation

- Design the Library of functions: May 31 - Jul 17
- Implement the Library (mid evaluation): Jul 18 - Aug 6
- Implement the Library (finalevaluation): Aug 7 – Oct 6

Library Testing and Reimplementation

- Evaluate the functions: Oct 6 - Oct 16
- Reimplementation of the Library: Oct 17 - Oct 26

Completion and Final Report

- Final Report: Aug 10 – Oct 26
- Final Presentation: Oct 27 – Nov 16

Total Duration: Dec 10 – Nov 16

Summary

- GPGPU is very common in modern computing.
- Execution time may be more precious than utilizing resources efficiently for some applications of computing.
- It has no directly related previous arts.
- Methodology
  - Characterize computations using some features.
  - Define the benchmarks, determining the boundary points of switching.
  - Implement Library of functions to be used by the programmer.