

Parallelising Kiwiplan GUI Framework

Mid Semester Presentation

Student – Prabhjot Singh

Overview

- Introduction
- Requirements
- Filter Details
- Parallelising basic filters
- Questions

Introduction

- Kiwiplan displays the information to users in the form of tables. User can customize the tables as per their requirements.
- When the user makes some operation on GUI, the data goes through certain sequence of filters.

Introduction

- Each filter does one particular task. Some of the common filters are sorting filters, grouping filters.
- There are around 40 different filters and each filter is independent of other filters.
- For some operations, the data goes through a series of filters. The order of filters are important to get the desired output.

Requirements

- All of the algorithms inside filters are being coded normally (Sequential programming).
- Due to the increasing size of tables, some of the operations take too long to complete and hence slows down GUI.
- Requirement is to investigate what steps of the algorithms can be done parallelly. That can lead to enormous speed ups.

Surprising Results

- In one of my other paper CS773 we were given 500 pairs of images and had to implement stereo matching algorithm.
- The sequential algorithm took 12500 seconds (3h 28m 20 sec) to run while when the same algorithm implemented parallelly it took only 750 seconds (0h 12m 30 sec).

Filter Details

- One of the most basic filter is the Sorting Filter.
- Shuttle sort is used to do the Sorting. It is a recursive algorithm.
- Sorts the items in a very similar way as Quick sort does, but shuttle sort is in-stable.

Sorting Filter

- Since Shuttle sort is based on divide-and-conquer nature, it can easily be parallelized.
- However, it is little bit better to sort the items using Quick sort and can be made stable depending on how the pivot is being handled.
- Sequential Quick sort is faster than Shuttle sort.

Sorting Filter

- Parallelising Quick sort does not require any synchronization. New thread starts as soon as a sub list is available.
- One thread works independently of other threads and does not need to communicate with others.
- Experimental results shown that parallelised version of Quick sort is indeed very efficient and can possibly replace Shuttle sort.

Other Filters

- Some of the other filters like Grouping filters, user expression validation filters algorithms also have some steps in which parallel programming can be used.
- Grand total filters does the task of computing the addition of various items. Again this is something which can be parallelised as well.

Current Status

- Hand in the Project plan to Hao.
- Decided the scope of the project.
- Finished reading some of the key concepts in parallel programming.
- Wrote couple of basic programs in parallel to see how much efficiency difference can be made.
- Did some research on the features introduced in Java 7.

Future Plan

- Immediate future plan is to parallelise the basic filters first.
- Plan is to make a standalone application which works independent of Kiwiplan Framework.
- Integrate them with the existing Kiwiplan products and thoroughly test it.
- Explore the new features of parallel computing in more detail introduced in Java 7.

Conclusion

- The progress is quite slow, but expected to speed up during holidays.
- Some experiments showed positive signs and it is indeed better to incorporate the features of parallel programming.
- Sticking to simple requirements makes the project more manageable.
- Parallel programming seems to be the future of programming.

Questions?