



Applying Parallelism in Kiwiplan Gui Framework

BTECH 450 PROJECT PLAN

PRABHJOT SINGH JASSAL

Project Description

Kiwiplan displays the information to the user in the form of tables known as flexible tables. The term flexible is used because users can view the information as per their requirements. When certain operation is performed by the user at a GUI level, the data goes through some sequence of filters, where each filter is doing just one task. Each of the algorithms in the filters is written in a manner which makes use of a single core to get the desired output (i.e. runs in a sequential order).

Due to the increasing size of tables, some of the operations takes too long to run and slows down GUI. Kiwiplan wishes to investigate if the efficiency of some of those operations can be increased by parallelising some parts of them.

Project Goal

The primary goal is to parallelise the algorithms within the filters by making use of new features introduced in Java 7 and integrate them with the existing Kiwiplan GUI framework. The secondary goal is to investigate how other products of Kiwiplan can be benefitted by using the similar ideas.

Different phases of the project

The project will be developed according to the software development lifecycle. The project is divided into 5 different phases.

- *Phase 1.* Familiar myself with the project.
- *Phase 2.* Design/Analysis.
- *Phase 3.* Implement the stand alone algorithms / prototype.
- *Phase 4.* Integrate the algorithms with the GUI framework.
- *Phase 5.* Testing / Re-work.

To minimize the errors, the phase 3 and 4 will be done for one filter first before moving onto the next filter. There are around forty different filters and it needs to figure out which parts can run in parallel and which one cannot.

The detailed description of what needs to be done at each phase is given next.

Phase 1. Familiar myself with the project.		
Proposed Tasks	Deliverables	Estimated Duration
1. Read the project outline and understand the goal and scope of the project.	N/A	2 Days
2. Understand the basic structure of filters. What task is accomplished by each filter and their expected input and output formats.	Make a draft and verify from Tim Walker.	4 Days
3. Understand the basic structure of pipelines - How the filters are connected to each other and in what order.	N/A	2 Days
4. Identify which filters are more inefficient and identify the possible bottleneck in them.	Make a draft and verify from Tim Walker.	3 Days
5. Make a project website.	Project Website	1 Day
6. Understand the basic terminology of the parallel programming and write the basic algorithms in parallel.	N/A	1 week

Phase 2, 3 and 4 will be done for one filter first before starting the next filter.

Phase 2. Design / Analysis		
Proposed Tasks	Deliverables	Estimated Duration
1. Identify what features of parallel programming can be used to make the selected filters more efficient.	Make a draft and verify that from Hao, Tim and Radu.	2 weeks

Phase 3. Implement the standalone algorithms		
Proposed Tasks	Deliverables	Estimated Duration
1. Implement the algorithms standalone so that it works correctly. The implementation will be independent of the Kiwiplan framework. The input and output format must be matched.	Working standalone algorithms and measure the clock time difference between the sequential and parallel algorithm.	1 month

Phase 4. Integrate the algorithms with the Kiwiplan framework.		
Proposed Tasks	Deliverables	Estimated Duration
1. Draft of how the code will be integrated. Create rough UML diagrams to show the connection.	Verify the draft that from Tim and Hao.	1 week
2. Integrate the algorithms with the Kiwiplan framework.	Fully functional efficient algorithms integrated with framework	1 month

Phase 5. Testing / Re-work.		
Proposed Tasks	Deliverables	Estimated Duration
1. Run the code on the existing test cases to see if the integration caused any damage.	Pass the JUnit test cases.	3 days.
2. Prepare final report	Btech project final report.	1 month.
3. Prepare final presentation.	Final presentation.	2 days.