Faculty of Computing, Engineering and Technology

STAFFORDSHIRE UNIVERSITY

Module Name: Task-Based Software Engineering

Module Number: COSE50584

Title of Assignment: Data Parallelisation

Module Learning Outcomes for This Assignment

1. DESIGN AND IMPLEMENT AN EFFICIENT TASK-BASED SOLUTION FOR A GIVEN SOFTWARE ENGINEERING PROBLEM THAT USES PARALLELISM.	Application
	Problem Solving

2. DEMONSTRATE A CRITICAL UNDERSTANDING OF THE KEY FEATURES OF PARALLEL TASK-BASED SOLUTIONS	Analysis
	Communication

3. ANALYSE, DEBUG AND ENHANCE THE EFFICIENCY OF A TASK-BASED PROGRAM THAT USES PARALLELISM	Application
	Knowledge & Understanding
	Problem Solving

Deadlines

Hand in deadline: Monday 9th April 2018 before 3:30pm

Assignment description

This assignment is worth 70% of the total module mark.

You are required to develop, demonstrate, and show improvements to an application making use of the Task Parallel Library (TPL). You will be expected to submit via Blackboard by the deadline given above.

More details of the assignment task are given in the following pages.

Assessment criteria

A detailed marking scheme is given at the end of this document.

Spur Ltd are a company that consists of 100 convenience stores in various locations around the UK. The have a modern Point of Sale system however their Supply Ordering system is a little out dated and they feel parts of their ordering process can be improved. Currently stores make orders with their suppliers independently of each other, this causes an issue for the finances team at Spur Headquarters as they find it very hard to collect all the information for the supplier orders made from the independent stores.

Each independent store generates numerous .csv files (1 a week) that contain the data for all the orders they make. The finances team would like a fast and responsive system that will allow them to analyse the data sent from the stores.

Spur Ltd are happy for you to experiment with solutions around this but have suggested that the use of the Task Parallel library would be beneficial as they believe it will impact the efficiency of the algorithm most effectively.

The finance team already have a method of receiving the files from the stores, they get placed into a central folder which they hope this new application will be able to read from. You will be supplied with 2 years of historical order data for each of the 100 stores to help you test your solution.

Minimum Requirements

- An easy to use command line interface.
- All features should be calculated when the option is selected, not at initial load. This will allow the data sets to be updated and the calculations rerun without reopening the application. Spur's technical team would like the UI to continue being responsive during these calculations.
- Have the flexibility to point the application to a certain folder on a PC to find the .csv files.
- List all stores, suppliers and supplier types.
- Allow the Finances team to find the following data:-
    - The total cost of all orders available in the supplied data
    - The total cost of all orders for a single store
    - The total cost of orders in a week for all stores
    - The total cost of orders in a week for a single store
    - The total cost of all orders to a supplier
    - The cost of all orders from a supplier type
    - The cost of orders in a week for a supplier type
    - The cost of orders for a supplier type for a store
    - The cost of orders in a week for a supplier type for a store

Advanced Requirements

- A Graphical User Interface using WinForms or WPF
- The ability to plot the historical supplier order data on a graph
    - Spur do not mind if a 3rd Party Graph control is used, some suggestions are below
- Other features the developer feels suitable may also be rewarded

Graph Control Options

MSChart (WinForms): https://code.msdn.microsoft.com/mschart

WPF Toolkit: http://wpf.codeplex.com/

Data File Specification

StoreCodes.csv lists all stores and their corresponding Store Codes.

File naming format: [Store Code]_[Week Number]_[Year].csv

File naming example: STA1_1_2013.csv
(Stafford Store for Week 1 of 2013)

File CSV format: [Supplier Name],[Supplier Type],[Cost of Order]

File CSV Example: Heinz,Beauty,103.53

Alternative Solution

To experiment with alternative solutions, you are also to cast the problem you have solved using C# into the F# language. You will also be required to make a detailed comparison in terms of language use and efficiency of solutions within the documentation you produce.

Documentation

As mentioned above, Spur also require documentation to be produced that details the design of the application. This should include information as to how the system has been designed with appropriate diagrams and detailed technical discussion around the system architecture and parallel programming concepts that have been applied in this system. To reiterate, this is to include your F# solution.

Performance Profiling and Evaluation

Spur technical team are also interested in the documentation of any performance related improvements and the expected speedup that can be achieved using the Task Parallel Library. Performance testing as well as an evaluation of the results would be very beneficial to the Spur Technical Team. You may wish to run tests of your application running sequentially to achieve accurate results.

What to submit

- A zip of the Visual Studio solution.
- A report including the System Design and Performance Evaluation.

How to Submit

Submission will be done via Blackboard. You should upload items separately but keep them within one submission on Blackboard. If you are unsure how to do this, please ask your tutor.

Marking scheme

The marks scheme is shown overleaf.

Feedback

Provisional assignment marks with written feedback will be posted on Blackboard by 7th May 2018 (20 working days).

Marking Scheme (Total Marks: 100)

Component		Available Marks	Criteria
Core Application Requirements (20 Marks)	Appropriate User Interface	5	An appropriate user interface that allows the operator to control the system
	File Selection	3	Allow the operator to select where to locate the .csv files
	List Details	2	The system can list all stores, suppliers and supplier types.
	Ability to show required data using TPL	10	1 Mark for ability to display results 1 Mark for each successfully implemented data look up as requested by the finance team
Quality (15 Marks)	Software Engineering	15	Technical Merit of Implementation (Use of TPL, Appropriate techniques, Level of functionality and technical quality)
Additional (25 Marks)	Improved UI	5	Improvements to UI (WinForms or WPF Application)
	Graphing Capabilities	10	The ability to render suitable graphs for the finance data
	Extra Features	10	Additional suitable features (Bonus marks for additional features using TPL)
F# Solution (20 Marks)		20	One you have completed your solution in C#, you are to cast it into F#. Once this is done you should, as part of your overall documentation, give a comparative critique of language features and effects on efficiency.
Report (20 Marks)		20	A report detailing the System Design (including appropriate diagrams) and a Performance Evaluation which includes improvements to your design for efficiency.