Martin Hawes
10238908

Project Title:

The Work-Stealing scheduling technique applied to computing the Mandelbrot Set.

Project Summary (same as on karl profile):

A concurrent algorithm which computes the mandelbrot set using a work-stealing scheduling technique.

Project Description:

The project involves an investigation into the effectiveness of the work-stealing technique when concurrently computing the Mandelbrot Set as an example case.

Work-stealing is a run-time scheduling technique in which a set of processing nodes (threads) have a queue of tasks to complete. When a processing node completes its queue, it actively looks to other processing nodes to "steal" tasks waiting in their queues. It is particularly useful when workload is unbalanced and can reduce the amount of time a processor remains redundant.

The Mandelbrot Set is a fractal which is ideal for computing in parallel because each point is independent from any other point.

I plan to produce an algorithm which computes the Mandelbrot Set using the techniques described above and discuss its effectiveness compared to other techniques, and identify the advantages and disadvantages.

Tasks, deliverables and demonstration:

- Produce a document detailing my research into and an explanation of the technique.
- Produce a document outlining research into potential tools to use. i.e. language features, libraries, techniques
- Produce a document explaining the Mandelbrot Set, it's properties, it's history, and why it is a suitable choice for my project.
- Produce a document outlining the design of the algorithm, detailing the design decisions made based on the research.
- Produce an algorithm which computes the Mandelbrot Set (not concurrent)
- Produce an algorithm which concurrently computes the Mandelbrot Set using work-stealing.
- Produce a detailed analysis of the effectiveness of the algorithm.
- Reflect on the project as a whole. Suggest ideas on scalability, expansion, alternatives, etc.

To demonstrate, my program will output a visual representation of the results. It could also implement a 'trace mode' in which it outputs the activity of each processing node (e.g. when a work-steal occurs) and record other interesting statistics (e.g. how many tasks a given node completes).

Martin Hawes
10238908

Background to the project

The program will be written in C++ as it provides excellent library support for concurrent programming and more low-level control than the likes of java. It is partially intended to repeat and verify some of the work carried out by Jason McGuiness in his masters thesis entitled 'Automatic Code-Generation Techniques for Micro-Threaded RISC Architectures'.

Following is a list of resources I used to decide to undertake this project:

- McGuiness, J. 2006. *Automatic Code-Generation Techniques for Micro-Threaded RISC Architectures.* MSc. Hertfordshire: University of Hertfordshire
- Janjic,V. 2011. *Load Balancing of Irregular Parallel Applications on Heterogeneous Computing Environments.* Ph. D. University Of St Andrews.
- Seinstra,F.J.;Maassen,J.;van Nieuwpoort,R.V.;Drost,N.;van Kessel,T.;van Werkhoven,B.;Urbani,J.;Jacobs,C.;Kielmann,T.;Bal,H.E. 2011 *Jungle Computing: Distributed Supercomputing beyond Clusters, Grids, and Clouds*. Amsterdam: Vrije Universiteit.
- Chase,D.;Lev,Y. 2005. *Dynamic Circular WorkStealingDeque.* Burlington: Sun Microsystems Laboratories; Burlington: Brown University.
- Blumofe,R.D.;Leiserson,E. 1999. *Scheduling Multithreaded Computations by Work Stealing.* Texas:The University At Texas Austin; Massachusetts: MIT Laboratory for Computer Science.
- Arora,N.S.;Blumofe,R.D.;Plaxton,C.G. 1998. *Thread Scheduling for Multiprogrammed Multiprocessors.* Texas:The University At Texas Austin.
- St Andrews University CS, 2012. *SCALES Work-Stealing Simulator.* [online] Available at:<http://www.cs.st-andrews.ac.uk/~jv/scales.html> [Accessed 2 November 2012].