

Design And Analysis Of A Plastic Parallel Programming System

Mark Jenkins



MInf Project (Part 2) Report

Master of Informatics

School of Informatics

University of Edinburgh

2018

Abstract

This is the scnd part of an MInf project which has spaned two years. In this project, we investigate the performance implications of plastic parallel programs which adapt to the current state of the host system, and explicitly cooperate to share and optimize the use of system resources.

In the first year of this project, we created an example of a parallel programming library which implemented these features, and gained some promising preliminary results into it's performance. In this second year, we develop upon this further by investigating a more insteresting parallel pattern...EXPAND

Contents

1	Introduction	3
1.1	3
2	Background And Previous Work	4
2.1	4
3	Design	5
3.1	Interesting Use Cases - Reasoning	5
3.2	Finding Interesting Use Cases	6
3.2.1	First	6
3.2.2	Second	6
3.2.3	Third	7
3.3	Selected Use Cases	7
4	Implementation	8
4.1	8
5	Experimental Methodology And Program	9
5.1	9

6	Results	10
6.1	10
7	Future Work And Conclusions	11
7.1	11
8	Bibliography	12

Chapter 1

Introduction

1.1

Chapter 2

Background And Previous Work

2.1

Chapter 3

Design

3.1 Interesting Use Cases - Reasoning

In order to be able to do our investigations, we need to select a couple of interesting use cases of stencil code.

The potential for experiments is massive, due to the large amount of variables we have access to. In our investigation, we have access to the following:

EXPAND UPON THESE

1. Number of iterations
2. Thread pinning
3. Combinations of multiple different workloads
4. Grid size
5. Number of workers

All of these can be changed from stage to stage. Beyond these parameters, we can have many different combinations of programs running simultaneously.

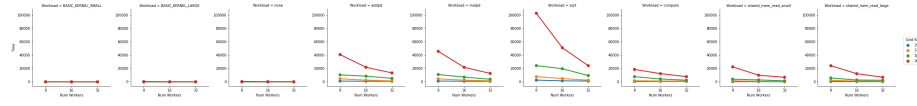
For the purposes of this project, we will select a few interesting use cases of the stencil pattern, and focus on testing these in a few different configurations.

3.2 Finding Interesting Use Cases

In order to find interesting use cases, we performed an initial foray into the experiment space we have access to.

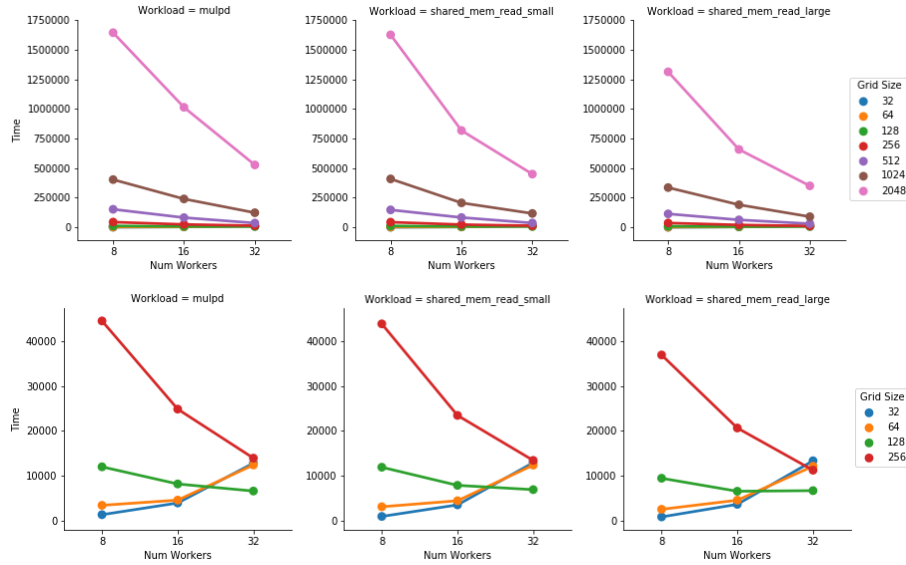
3.2.1 First

Firstly we compared the performance characteristics of each of our different workloads, and how they performed with different numbers of threads.



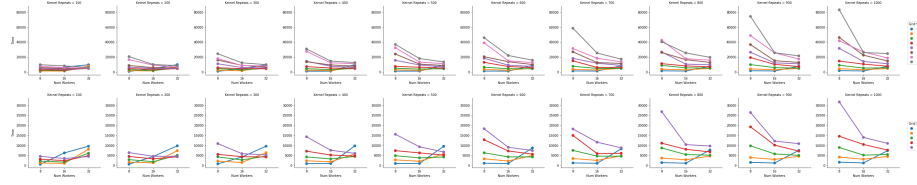
3.2.2 Second

Next we picked some interesting kernels from the previous experiment, and investigated them further, with some smaller grid sizes. This was because we were interested in a use case where at a certain point, adding more threads is no longer optimal.



3.2.3 Third

Next, we focused on the mulpd kernel, and tested a variety of kernel repeats, and smoe more fine grained grid sizes. This is so that we could identify the point at which, as the grid size shrinks, increasing the number of threads will be sub-optimal.



As you can see from the graphs, for gridsizes 32-128, going beyond 16 threads did not increase performance in all cases.

Another interesting take from the graphs is that we can see that as the grid size increases, the runtime curves 'flip' at 32 threads such that smaller grid sizes take longer. However, as the number of kernel repeats increases, this flipping gradually lessens. This is because the increased independent workload per thread starts to

3.3 Selected Use Cases

Chapter 4

Implementation

4.1

Chapter 5

Experimental Methodology And Program

5.1

Chapter 6

Results

6.1

Chapter 7

Future Work And Conclusions

7.1

Chapter 8

Bibliography