A parallelised and distributed approach to implementing Conway's Game of Life

Aaron Chan
University of Bristol
ho21739@bristol.ac.uk

Ferdinand Hubbard
University of Bristol
ej21378@bristol.ac.uk

December 4, 2022

Abstract

We intend to explain our two solutions for implementing Conway's Game of Life in Golang, namely the Parallel and Distributed versions. Within this report, we will discuss the impacts of thread usage in both our implementations, as well as the effect of distributed computation as opposed to single-machine computation. In essence, we found that execution time was sped up as we increased thread count, reduced branch instructions and implemented a halo exchange mechanism.

I. Introduction to the parallel solution

As an initial attempt, we began by implementing a single-threaded Game of Life (GoL) Engine, expanding upon this work with parallelisation by virtue of concurrent go-routine workers. Recognising that there exists various methods to calculating the perimeter for `getNeighbourCount`, we tried various counting methods - taking into account the wrap-around feature of Game of Life.

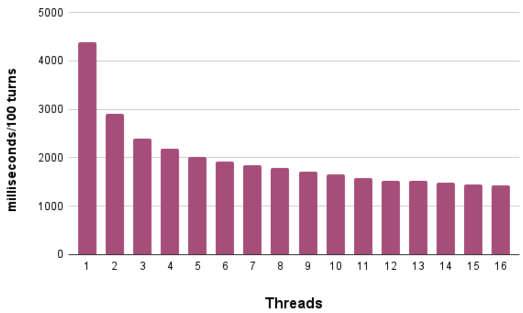

Figure 1: *Modulo Implementation*

II. Benchmarking methodology

Our means of evaluating parallelisation performance gains was via the Go testing framework. Using 512px x 512px PGM images, we repeated each benchmark 10 times, taking the average runtime for each thread to acquire the central value, and plotting them. Each feature implementation was benchmarked in the same fashion.

III. Parallelising the program

The baseline implementation's distributor function starts with reading the image into 2D slice via *io* commands, before evenly dividing the workload between workers. With the aid of go-routines, workers have access to the old slice as well as the boundaries in which to process. By using `getNeighbourCount`, we apply the Game of Life laws onto the world, piping results into a blank slice of size:

$$(endRow - startRow) \times ImageWidth$$

Sending the results back to the distributor, we integrate back together the new slices, move the pointer to the old world to this newly-merged world and send the appropriate signals on the *events*, *io* channels.

- show workers pseudocode
- show handler pseudocode

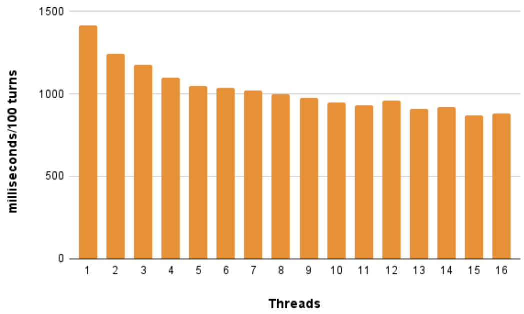

Figure 2: *Branching Implementation*

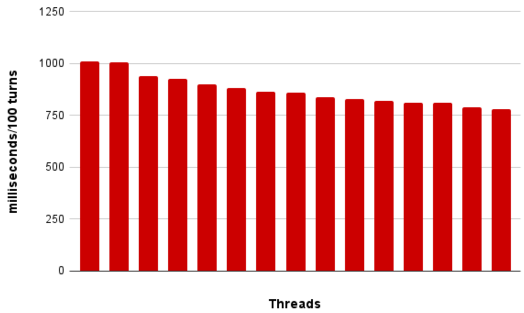

Figure 3: *Bitmask Implementation*

- flowchart/ diagram of how channel communication workers
- simple flowchart of how the whole program works currently

IV. On the topic of analysis

From Fig. 1, we can conclude that multithreading have an overall improvement on performance. Statistically-speaking, we find the solution using modulo operations is most affected by thread usage, having standard deviation 413ms, as opposed to 107ms and 73ms for branching and bitmasking respectively. Interested persons will find our results achieved by:

$$SD(\forall x \in runtime : max(runtime) - x)$$

This, however, implies not that higher threads means greater performance as there exists other, non-parallelisable tasks, such as reading PGM images set a base level of execution time.

- show graph for 5120x5120.gpm for single implementation vs parralel implementation
- describe the differences between each graph
- explain the differences by referencing the flowchart/ how the code worker
- explain the shape of the graph (why it resembles a $1/x$ graph). e.g. bottlenecks in channel communication...

V. Memory sharing

THIS SECTION WILL BE USED IF WE DECIDE TO IMPLEMENT MEMORY SHARING OVER CHANNEL COMMUNICATION

VI. Benchmarking and critical analysis of the memory sharing

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Etiam lobortis facilisis sem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi nec ante. Donec ullamcorper, felis non sodales commodo, lectus velit ultrices augue, a dignissim nibh lectus placerat pede. Vivamus nunc nunc, molestie ut, ultricies vel, semper in, velit. Ut porttitor. Praesent in sapien. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Duis fringilla tristique neque. Sed interdum libero ut metus. Pellentesque placerat. Nam rutrum augue a leo. Morbi sed elit sit amet ante lobortis sollicitudin. Praesent blandit blandit mauris. Praesent lectus tellus, aliquet aliquam, luctus a, egestas a, turpis. Mauris lacinia lorem sit amet ipsum. Nunc quis urna dictum turpis accumsan semper.

VII. Implementing Halo Exchange

TO BE IMPLEMENTED

VIII. Benchmarking and critical analysis of the Halo Exchange

Lorem ipsum dolor sit amet, consectetur adipiscing elit. TO BE IMPLEMENTED

IX. Benchmarking and critical analysis of enabling SDL

- show graphs and describe shape and differences of with vs without SDL
- explain how SDL works
- explain how this causes bottlenecks

X. Comparing the parralel benchmarks

Maecenas sed ultricies felis. Sed imperdiet dictum arcu a egestas.

- Donec dolor arcu, rutrum id molestie in, viverra sed diam
- Curabitur feugiat
- turpis sed auctor facilisis
- arcu eros accumsan lorem, at posuere mi diam sit amet tortor
- Fusce fermentum, mi sit amet euismod rutrum
- sem lorem molestie diam, iaculis aliquet sapien tortor non nisi
- Pellentesque bibendum pretium aliquet

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Etiam lobortis facilisis sem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi nec ante. Donec ullamcorper, felis non sodales commodo, lectus velit ultrices augue, a dignissim nibh lectus placerat pede. Vivamus nunc nunc, molestie ut, ultricies vel, semper in, velit. Ut porttitor. Praesent in sapien. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Duis fringilla tristique neque. Sed interdum libero ut metus. Pellentesque placerat. Nam rutrum augue a leo. Morbi sed elit sit amet ante lobortis sollicitudin. Praesent blandit blandit mauris. Praesent lectus tellus, aliquet aliquam, luctus a, egestas a, turpis. Mauris lacinia lorem sit amet ipsum. Nunc quis urna dictum turpis accumsan semper.

Text requiring further explanation[1].

XI. Results

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Etiam lobortis facilisis sem. Nullam nec mi et neque pharetra sollicitudin. Praesent

Table 1: *Example table*

Name		
First name	Last Name	Grade
John	Doe	7.5
Richard	Miles	2

imperdiet mi nec ante. Donec ullamcorper, felis non sodales commodo, lectus velit ultrices augue, a dignissim nibh lectus placerat pede. Vivamus nunc nunc, molestie ut, ultricies vel, semper in, velit. Ut porttitor. Praesent in sapien. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Duis fringilla tristique neque. Sed interdum libero ut metus. Pellentesque placerat. Nam rutrum augue a leo. Morbi sed elit sit amet ante lobortis sollicitudin. Praesent blandit blandit mauris. Praesent lectus tellus, aliquet aliquam, luctus a, egestas a, turpis. Mauris lacinia lorem sit amet ipsum. Nunc quis urna dictum turpis accumsan semper.

$$e = mc^2 \tag{1}$$

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Etiam lobortis facilisis sem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi nec ante. Donec ullamcorper, felis non sodales commodo, lectus velit ultrices augue, a dignissim nibh lectus placerat pede. Vivamus nunc nunc, molestie ut, ultricies vel, semper in, velit. Ut porttitor. Praesent in sapien. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Duis fringilla tristique neque. Sed interdum libero ut metus. Pellentesque placerat. Nam rutrum augue a leo. Morbi sed elit sit amet ante lobortis sollicitudin. Praesent blandit blandit mauris. Praesent lectus tellus, aliquet aliquam, luctus a, egestas a, turpis. Mauris lacinia lorem sit amet ipsum. Nunc quis urna dictum turpis accumsan semper.

XII. Discussion

i. Areas for improvement

A statement requiring citation [Figueredo and Wolf, 2009]. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Etiam lobortis facilisis sem. Nullam nec mi et

[1] Example footnote

neque pharetra sollicitudin. Praesent imperdiet mi nec ante. Donec ullamcorper, felis non sodales commodo, lectus velit ultrices augue, a dignissim nibh lectus placerat pede. Vivamus nunc nunc, molestie ut, ultricies vel, semper in, velit. Ut porttitor. Praesent in sapien. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Duis fringilla tristique neque. Sed interdum libero ut metus. Pellentesque placerat. Nam rutrum augue a leo. Morbi sed elit sit amet ante lobortis sollicitudin. Praesent blandit blandit mauris. Praesent lectus tellus, aliquet aliquam, luctus a, egestas a, turpis. Mauris lacinia lorem sit amet ipsum. Nunc quis urna dictum turpis accumsan semper.

ii. Thanks

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Etiam lobortis facilisis sem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi nec ante. Donec ullamcorper, felis non sodales commodo, lectus velit ultrices augue, a dignissim nibh lectus placerat pede. Vivamus nunc nunc, molestie ut, ultricies vel, semper in, velit. Ut porttitor. Praesent in sapien. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Duis fringilla tristique neque. Sed interdum libero ut metus. Pellentesque placerat. Nam rutrum augue a leo. Morbi sed elit sit amet ante lobortis sollicitudin. Praesent blandit blandit mauris. Praesent lectus tellus, aliquet aliquam, luctus a, egestas a, turpis. Mauris lacinia lorem sit amet ipsum. Nunc quis urna dictum turpis accumsan semper.

References

[Figueredo and Wolf, 2009] Figueredo, A. J. and Wolf, P. S. A. (2009). Assortative pairing and life history strategy - a cross-cultural study. *Human Nature*, 20:317–330.