MPI Assignment Report: Parallel Game of Life

Karim Bacchus (CID: 00942807)

February 23, 2020

1 Speedup and Efficiency of Parallelisation for $N \leq 16$

Throughout this report all our simulations occur with $1,000$ iterations of playing the Game of Life with periodic boundaries. We test the speedup for $N = 1, 2, 4, 8, 16$ of square grids.

In Figures 1 and 2 we can see there for all sizes the rate of speed up and efficiency declines for increasing N, and this is more pronounced for the smaller grid size 2000×2000 and even more so for 1000×1000. This is as expected; if the overall grid size of the Game is smaller, the subdivided grid each node is responsible for is also smaller, thus a greater percentage of time is spent updating the "ghost" edges of each subgrid (i.e. transferring data between each node) versus calculating an iteration of the Game of Life on each node.

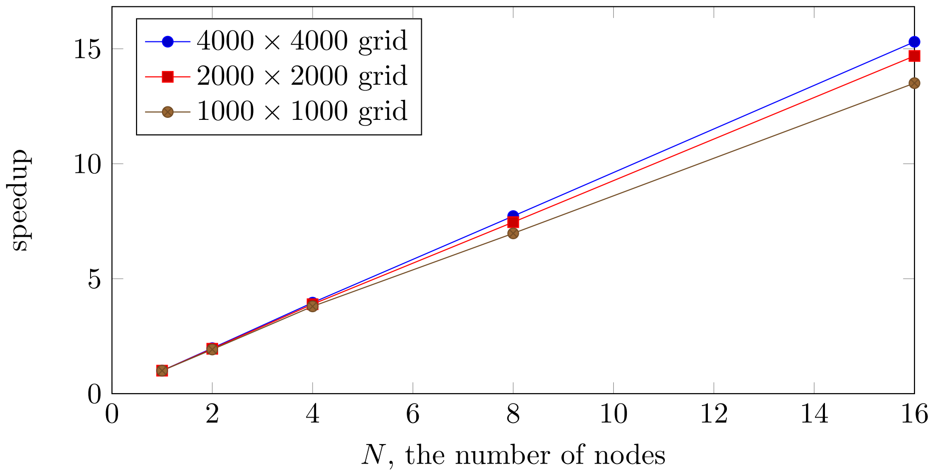

Figure 1: Speedup

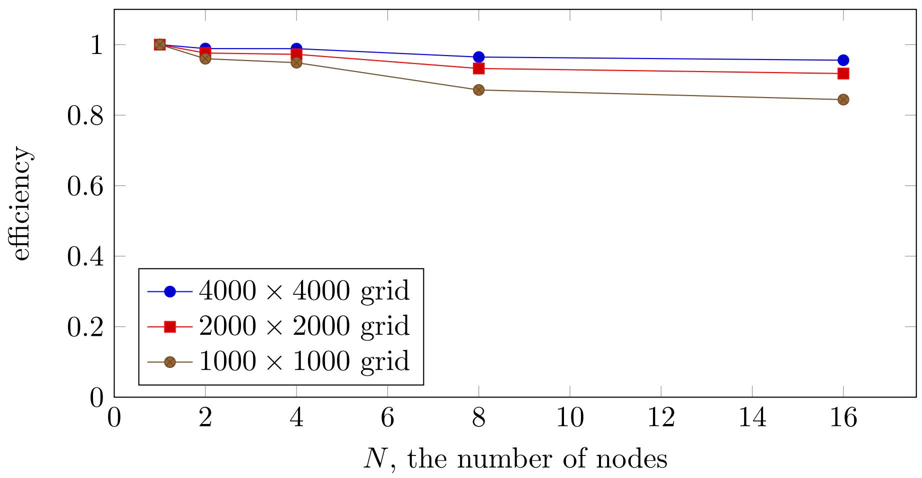

Figure 2: Efficiency of speedup

2 Speedup and Efficiency of Parallelisation for $N > 16$

We can see the clearer difference in how the rate of speedup and efficiency declines when using a much larger number of nodes; we plot below in Figures 3 and 4 additional points for when $N = 48, 72, 96, 120, 144, 168$.

We observe that for grid size 1000×1000 the rate of speedup dramatically reduces past $N = 48$ and flattens out to a speed up of around 52 (the dashed line). This corresponds to the efficiency dropping substantially at this point in Figure 4.

The 4000×4000 grid size only starts to see a noticeable drop in efficiency at $N = 168$, and it maintains an advantage compared to the 2000×2000 which increases as the number of nodes increases.

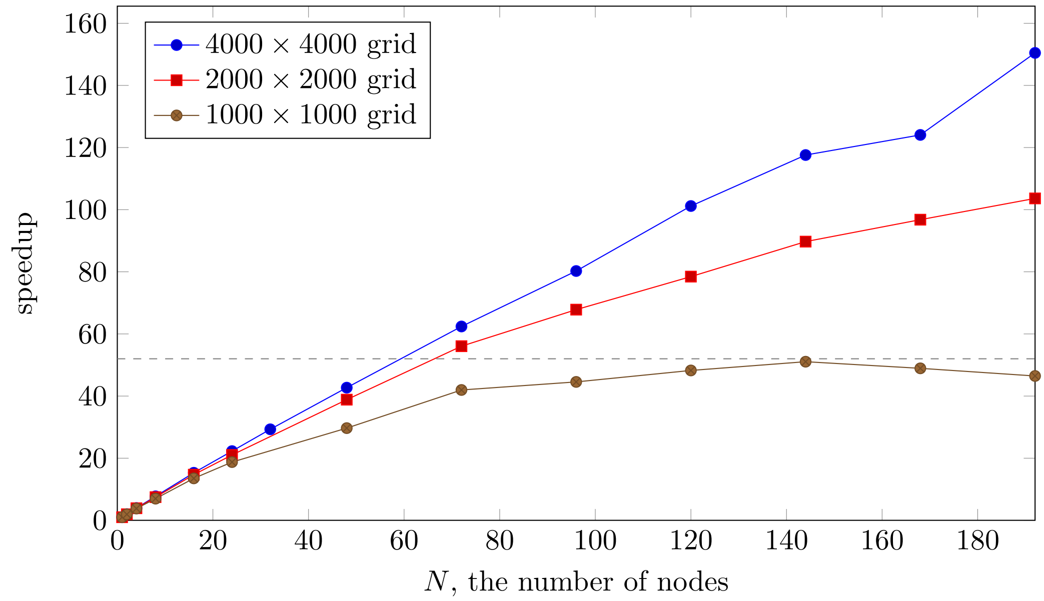

Figure 3: Speed up for large N

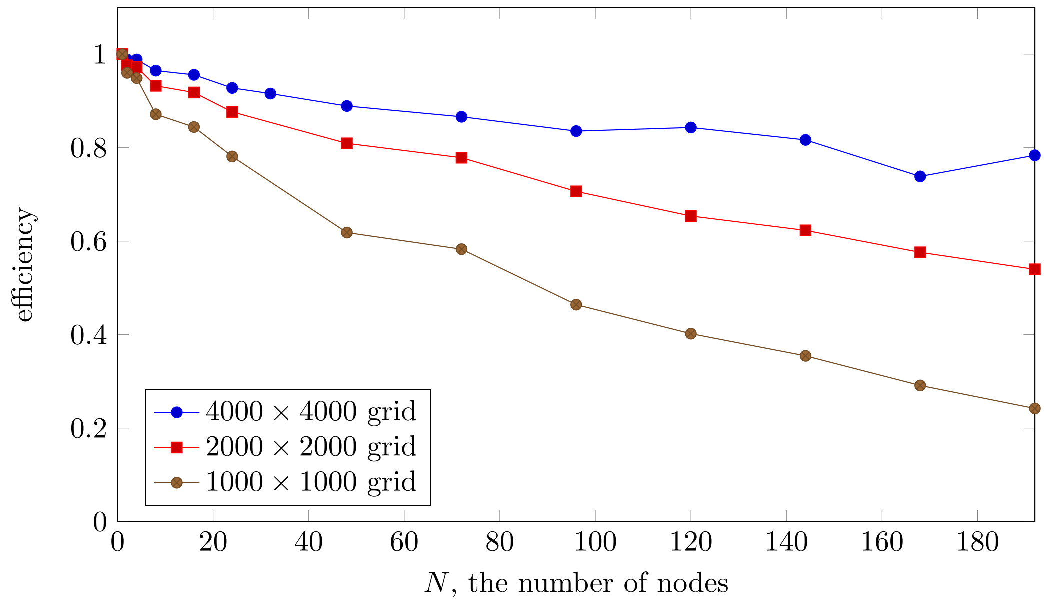

Figure 4: Efficiency of speedup for large N

2.1 Efficiency between different grid sizes

Theoretically for a 2D domain decomposition problem we'd expect the efficiency to be around

$$E \approx \frac{1}{1 + k(A/N)^{-1/2}}$$

In particular we would expect that the efficiency for a square grid with area A with N nodes would be roughly the same as the efficiency with a square grid of area kA with kN nodes.

In Figure 4 we see that the efficiency for $N = 192$ on the 2000×2000 grid is in fact noticably lower at 0.540 versus the the efficiency of the 1000×1000 grid for $N = 192/4 = 48$, which is 0.618. Similarly the efficiency for $N = 192$ on the 4000×4000 grid is 0.783 which is less than 0.809, the efficiency for $N = 48$ on the 2000×2000 grid.

This discrepancy is probably due to the increased overhead in communicating with an increased number of nodes that the approximation does not take into account, which is why the predicted efficiency is lower even though A/N remains constant in the comparisons above.

3 How does change in height and width change efficiency?

We plot the timings for various arrangements of a grid with area $1600 \times 1600 = 2560000$ in Figure 5 below. Clearly as we get further away from a square the timing increases; this is most extremely observed in the case with the "thinnest" domain, a 25×102400 grid:

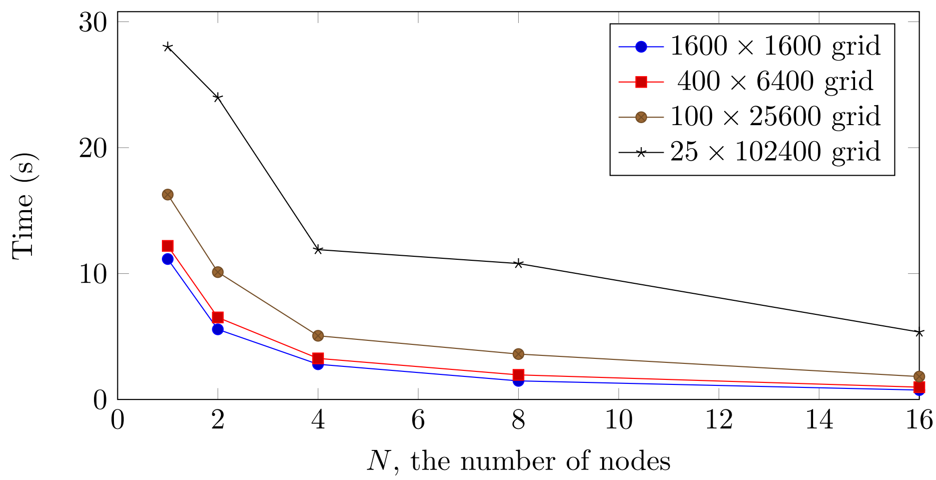

Figure 5: Timing with changing width of a fixed Area $A = 2560000$

This is to be expected; the area remains constant so the timing of each Game of Life calculation is the same for all cases, while as the grid becomes more rectangular, the surface area of each subgrid increases, increasing the time of communicating between each node and thus the overall time.

(Incidentally interchanging the width and height does not change timings as our method for dividing the grid into subgrids for each processor and then doing communication between the subgrids is symmetric.)

– End of Coursework –