Load Balancing

- How do we evenly distribute the work?
- Assigning work to processors statically
- In a 1D problem, like count3s:

![Diagram showing work distribution between two threads](thread0_thread1.png)
2D Problem
A stencil computation, such as needed for simulating changing temperature over a surface.
2D Problem
The Game of Life
by John Conway

Cellular Automaton:
Grid of cells, the behavior of each cell is governed by rules.
Rules are applied in generations (the state of a cell in generation \( t \) depends upon its state and that of its neighbors in generation \( t-1 \)).
Game of Life

- The grid of cells is a torus (i.e. it wraps around the sides and the top/bottom)

- Rules for a cell
  - If alive at generation $t-1$ and have 2 or 3 living neighbors, then remain alive in generation $t$
  - Otherwise, die.
  - If dead at generation $t-1$ and have 3 living neighbors, then come alive in generation $t$
  - Otherwise, stay dead
Game of Life

The grid of cells is a torus (i.e. it wraps around the sides and the top/bottom)

Rules for a cell

- If alive at generation $t-1$ and have 2 or 3 living neighbors, then remain alive in generation $t$
- Otherwise, die.
- If dead at generation $t-1$ and have 3 living neighbors, then come alive in generation $t$
- Otherwise, stay dead

Generation 0
Game of Life

- The grid of cells is a torus (i.e. it wraps around the sides and the top/bottom)

- Rules for a cell
  - If alive at generation t-1 and have 2 or 3 living neighbors, then remain alive in generation t
  - Otherwise, die.
  - If dead at generation t-1 and have 3 living neighbors, then come alive in generation t
  - Otherwise, stay dead

Generation 1
Game of Life

- The grid of cells is a torus (i.e. it wraps around the sides and the top/bottom)

- Rules for a cell
  - If alive at generation t-1 and have 2 or 3 living neighbors, then remain alive in generation t
  - Otherwise, die.
  - If dead at generation t-1 and have 3 living neighbors, then come alive in generation t
  - Otherwise, stay dead

Generation 0
Game of Life

The grid of cells is a torus (i.e. it wraps around the sides and the top/bottom)

Rules for a cell
- If alive at generation t-1 and have 2 or 3 living neighbors, then remain alive in generation t
- Otherwise, die.
- If dead at generation t-1 and have 3 living neighbors, then come alive in generation t
- Otherwise, stay dead

Generation 1
Game of Life: Sequential

- 2 boards
  - 1 is generation t-1
  - the other generation t

- Each board is a 2D integer array of 0’s and 1’s

- Fill in generation t’s board by applying rules to generation t-1’s board
Game of Life: Parallel

- Assign each processor a set of cells
- Coordinate among the processors to make sure all are working on the same generation at the same time
void barrier() {
    pthread_mutex_lock(&barrier_lock);
    count++;
    if(count==t) {
        count=0;
        pthread_cond_broadcast(&all_here);
    }
    else {
        pthread_cond_wait(&all_here, &barrier_lock);
    }
    pthread_mutex_unlock(&barrier_lock);
}