Simple SIR Infection Model in FLAME

David Worth, Chris Greenough, Shawn Chin

Software Engineering Group

Computational Science & Engineering Department
Rutherford Appleton Laboratory

david.worth@stfc.ac.uk, christopher.greenough@stfc.ac.uk, shawn.chin@stfc.ac.uk

Background

- Transfer NetLogo infection model to FLAME
- Agents move randomly on torroidal domain
- One of 3 states
  - Susceptible
  - Infected
  - Removed (immune)

- Birth and death included
- Spread controlled by
  - Infectiousness, chance of recovery, duration of virus

One Iteration

- Agent moves
  - 1 unit in direction ±100º of current heading
- Infected agents post location
- Susceptible agents read locations
  - Look for messages within their 1x1 patch
  - Calculate chance of becoming infected
  - Based on infectiousness
- Infected agents calculate chance of recovery
  - Based on duration of virus & chance of recovery
- Non-sick agents have chance of reproducing
  - Up to carrying capacity
  - Based on agent lifespan & average number of offspring

Implementation

- One **Person** agent
  - Agent identification: **Id**
  - Position: **x**, **y** (double) and **heading** (double)
  - State flags: **is_sick**, **is_immune**
  - Counters: **sick_count** (how long infected), **age** (how old)
- One **infected** message
  - Agent id: **Id**
  - Position: **x**, **y** (double)
- Functions
  - **get_older** (Start ⇒ 1)
  - **move** (1 ⇒ 2) Output **infected** message
  - **infect** (2 ⇒ 3) Input **infected** message
  - **recover** (3 ⇒ 4) Depends on **infect** function
  - **Reproduce** (4 ⇒ End) Depends on **recover** function

Agent Creation

- Required by **`reproduce`** function
- Need unique **`ids`**
- New agent created from existing one so use existing id as basis
  - Add on global number of agents * current iteration number
  - Increment global number of agents
- OK because agents only have one child per iteration
- Not complete solution
  - global number of agents changed by other functions

Science & Technology Facilities Council

Environment

- Fixed values defining: reproduction, disease, domain
  - Lifespan 100
  - Average offspring 4
  - Carrying capacity – scaled with initial number of agents
  - Infectiousness 65%
  - Chance of recovery 50%
  - Duration of disease 20
  - Domain height – scaled with initial number of agents
  - Domain width – scaled with initial number of agents

Science & Technology Facilities Council

Input Data

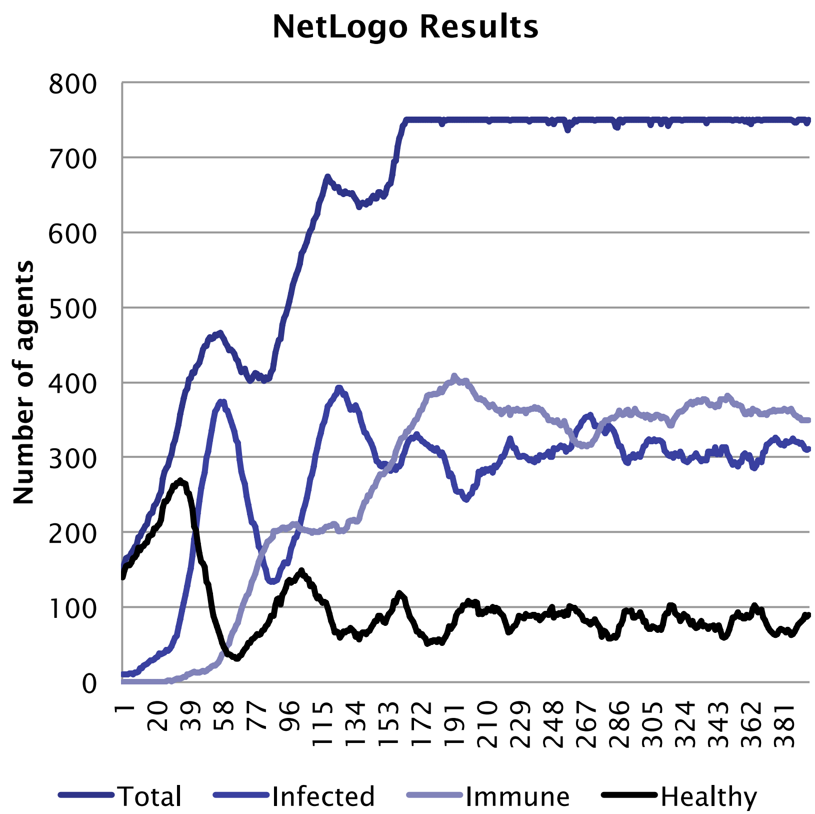
- Initially same as NetLogo model
  - 150 agents
  - 10 infected (choose first 10)
  - 34x34 domain
  - Carrying capacity = 750
  - Position and heading random uniform distribution

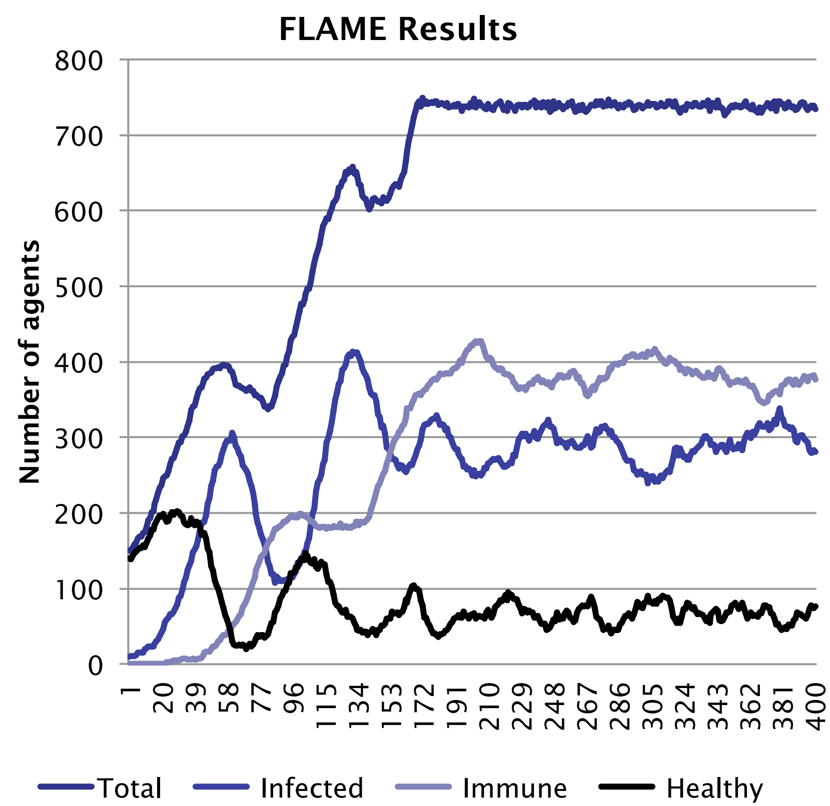
- Other values on previous slide

- Generated with Python script
  - `./init_start_state.py <width> <height> <agent_count>`
  - Scale domain with agent count to keep same density
  - Change carrying capacity in script!!

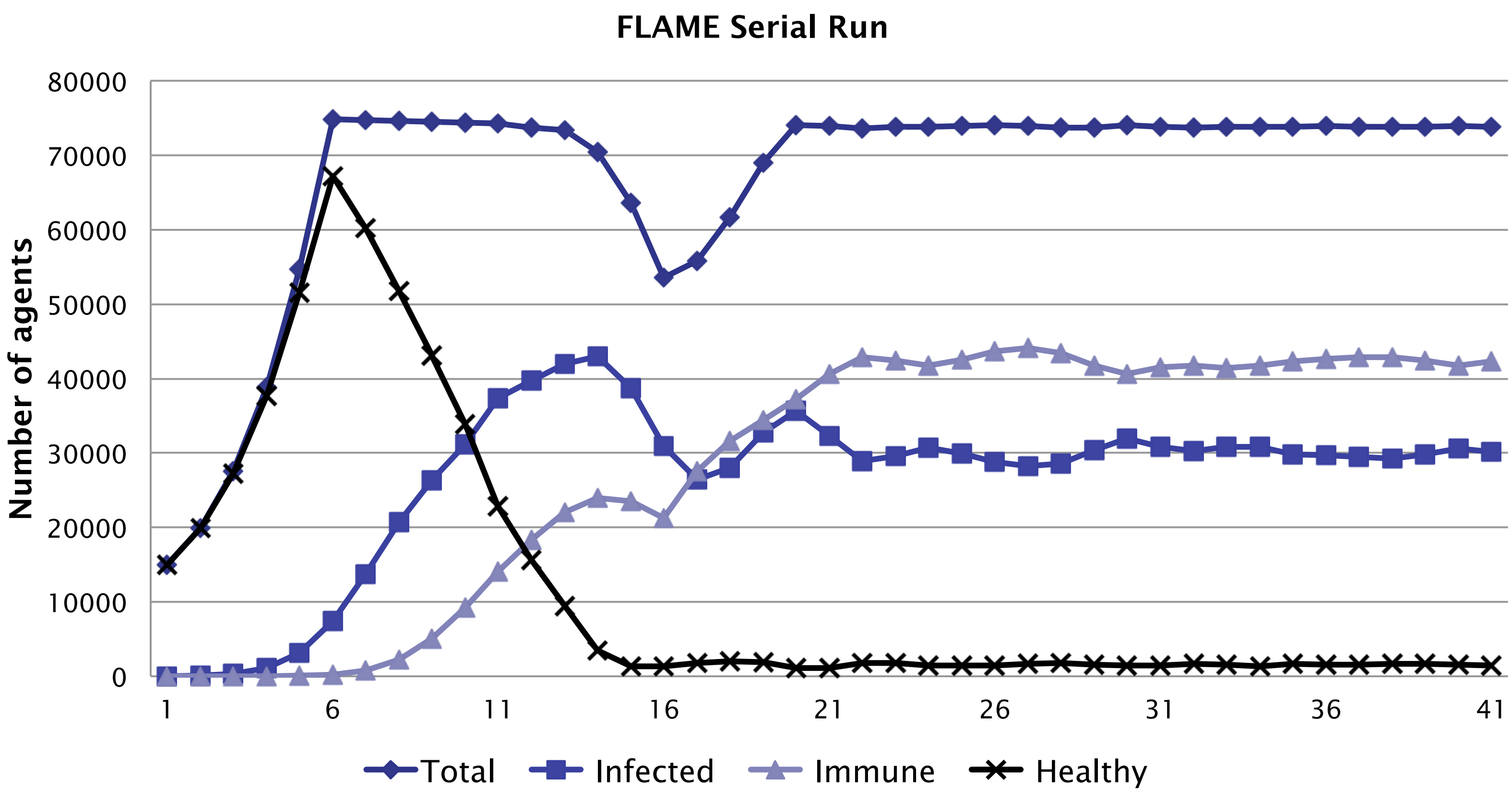
Verification

- Check with NetLogo





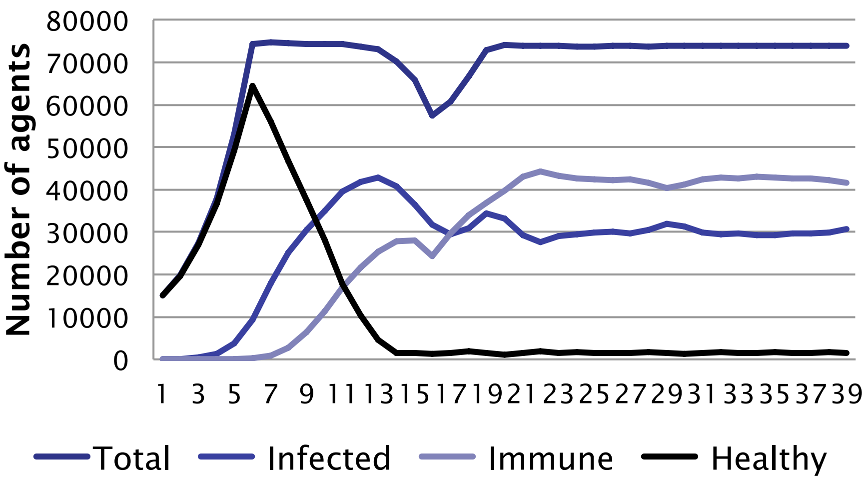
Serial run 15000 agents



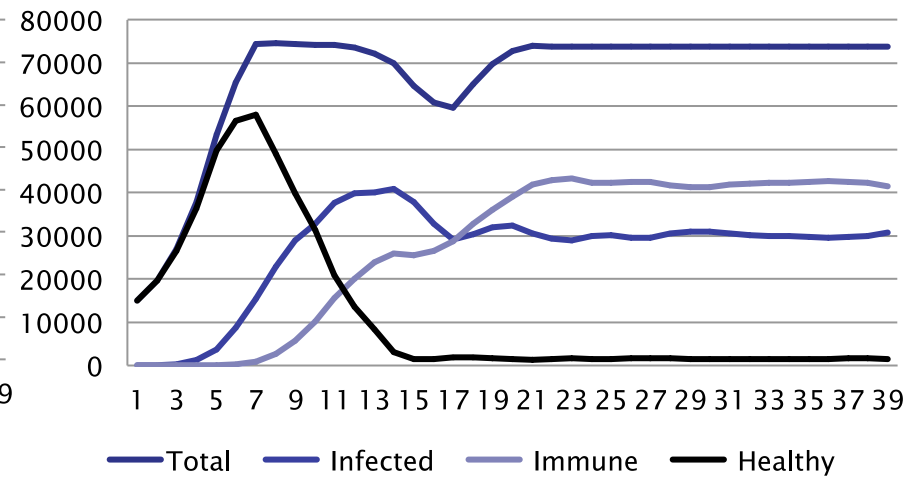
Parallel Runs

- Carrying capacity of domain is global data
  - Split capacity equally between nodes is only choice
  - Try to keep agent number same on all nodes therefore…
  - Do round-robin agent partitioning
  - Does give “better” results

Parallel - Round Robin

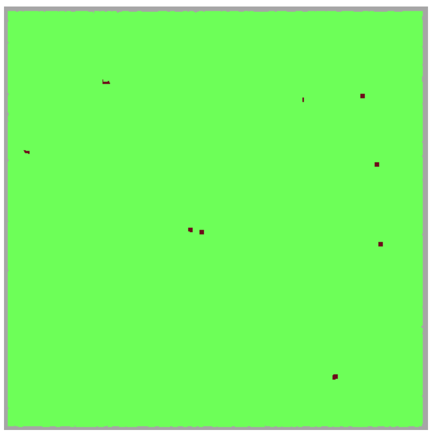


Parallel - Geometric

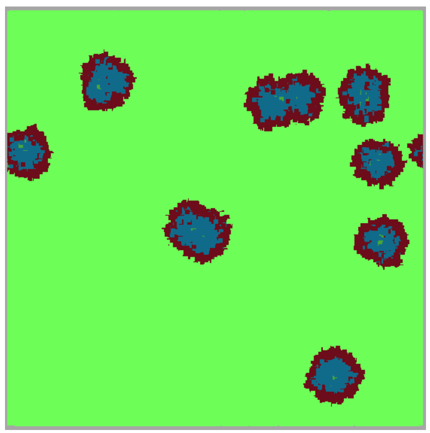


Pretty Pictures

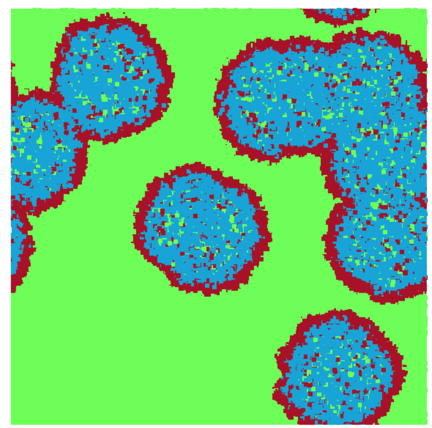
- Run on HECToR
- 500 cores
- 150000 initial agents
- 750000 carrying capacity



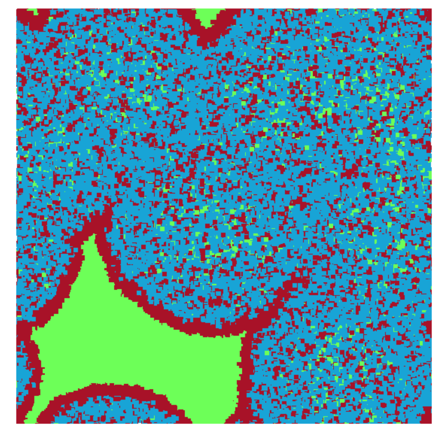
Iteration 10



Iteration 100



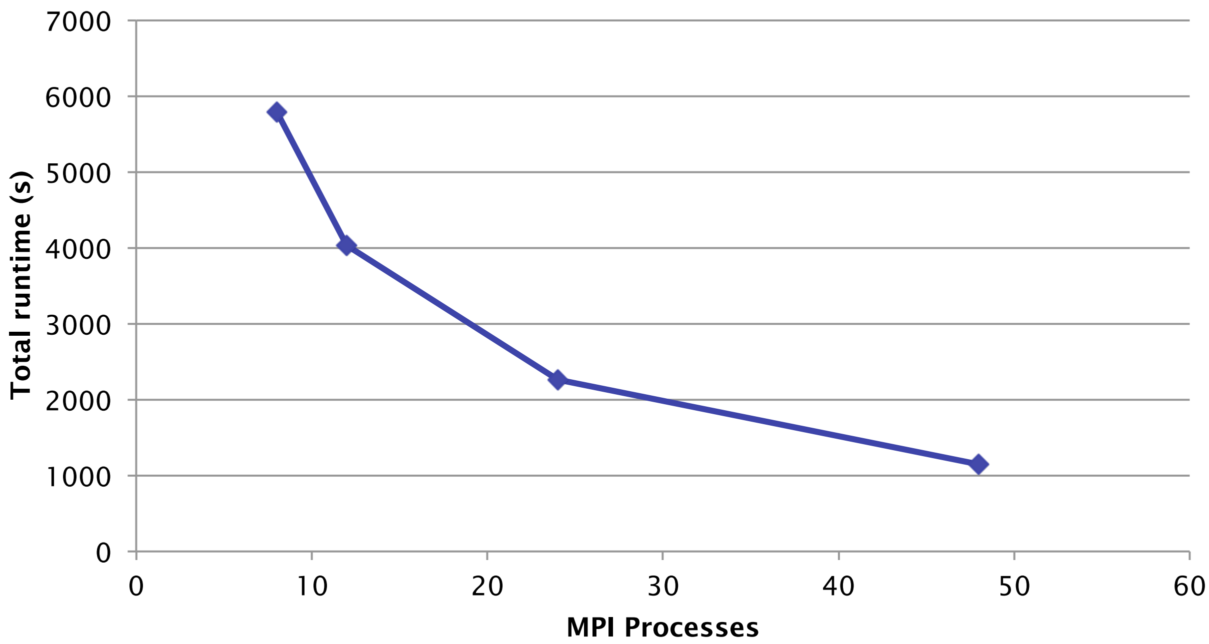
Iteration 200



Iteration 300

Timing data

- HECToR
- 15,000 agents
- Completely unreliable!



Science & Technology Facilities Council

Conclusions

- Improvements to FLAME
  - Global variables
    - Update frequency – every change, end of iteration, programmatic
    - Partition of values among nodes – e.g. carrying capacity
  - Geometric partitioning better for infection model if GVs available
    - Halo filters
  - Agent migration if using geometric partitioning

- NetLogo = bad model
  - Missing potential infection because of patches