DISEASE SPREAD SIMULATION

Sumanth Reddy Dasi (nagabrah@buffalo.edu)

Instructor: Dr. Russ Miller

CSE 633 – Parallel Algorithms

University at Buffalo The State University of New York

Agenda

- Disease Spread Models
- Scenario
- Need to Parallelize
- Simulation Steps
- Parallelization
- Observations
- Future Works
- References

Disease Spread Model

What?

- Mathematical modelling : modelling the underlying system to predict its behavior
- Outbreak in a population : in our case the system we are trying to model is the disease spread across individuals in a population

Disease Spread Model

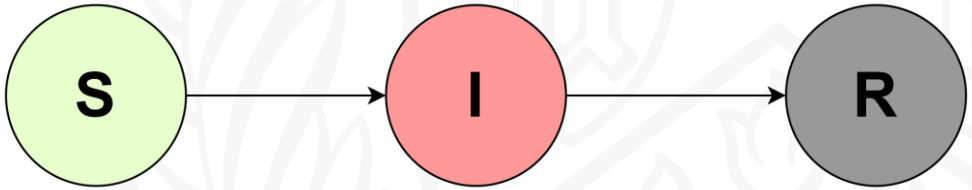
When?

- Predict the spread
- Evaluate control strategies
- Real time planning

The earliest account of mathematical modelling of spread of disease was carried out in 1760 by Daniel Bernoulli.

SIR Model

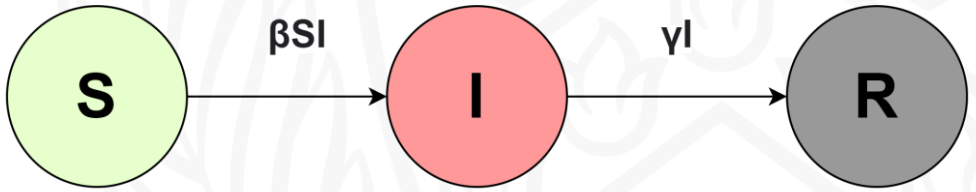
- Susceptible (S) : susceptible to the infection
- Infected (I) : infected and capable of spreading infection to others
- Removed (R) : recovered from infection and gained immunity



SIR Model

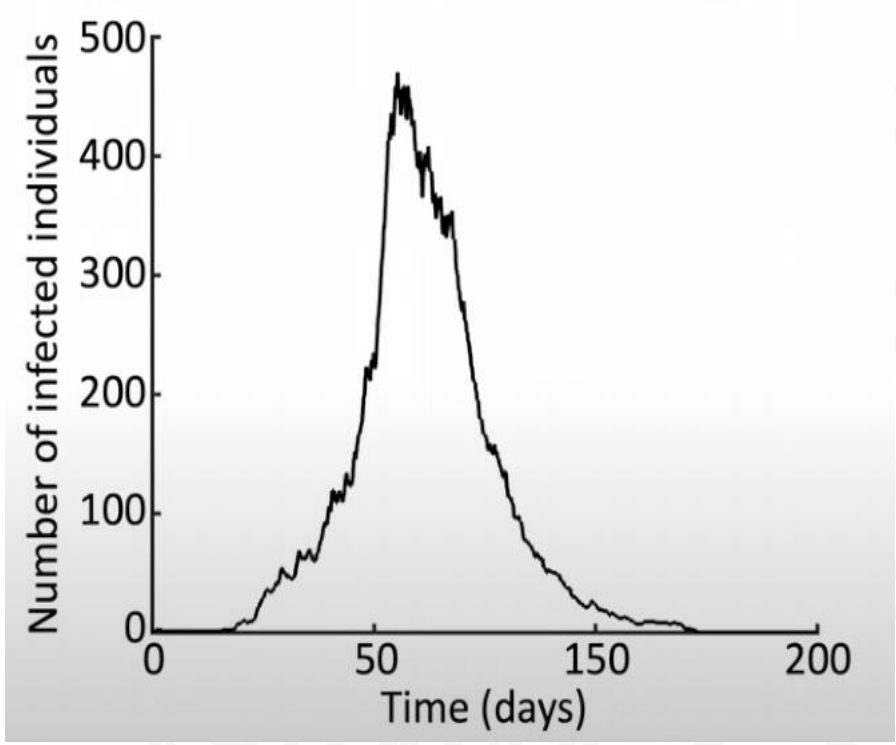
- Newly Infected : βSI
  - Proportional to S (targets)
  - Proportional to I (infected individuals)

- Newly Removed : γI
  - Proportional to I (infected individuals)



Extensions to model

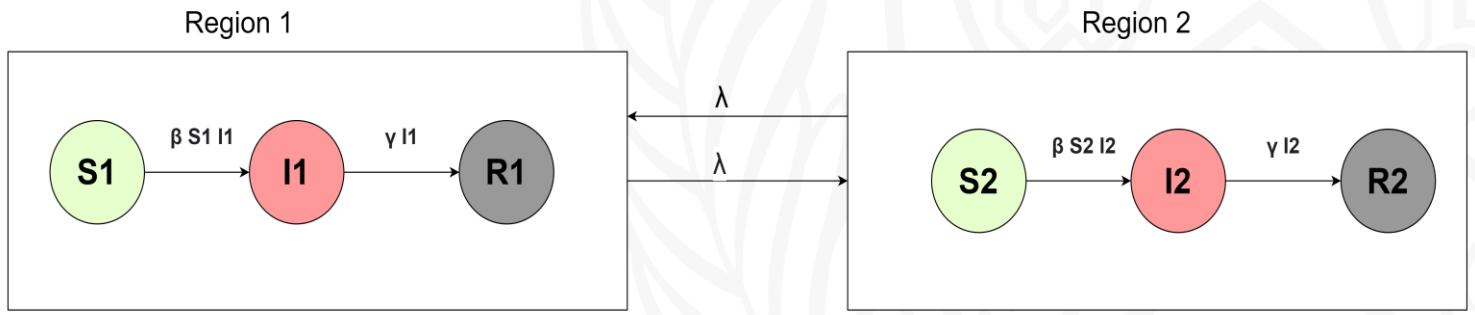
- **Stochastic model :** The initial model is deterministic, so we introduce stochasticity in infection spread to our model to make it closer to real world.



Source: Oxford Mathematics Public Lecture Robin Thompson link

Extensions to model

- Stochastic spread
- **Spatial structure :** In the real-world individuals are scattered across regions so we add this extension to the model. Moreover, different groups are differently connected with each other, so the infection moves across the groups based on how well they are connected.



Extensions to model

- Stochastic spread
- Spatial structure
- **Infected time :** Based on the epidemiology of the disease, different infections last for different days. SIR models can be extended to include this behavior.

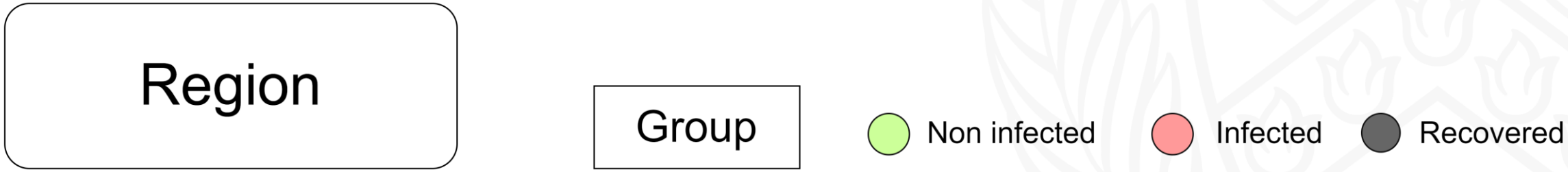
Need to Parallelize

- Faster results lead to faster safety measures
- Can help simulate scenarios with complex interactions

Simulation Scenario

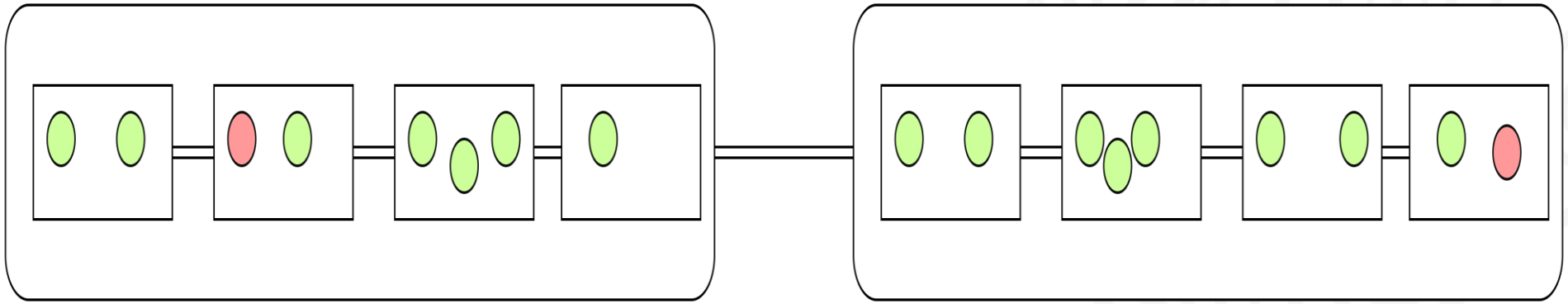
- Regions
- Groups
- Individuals

The simulation world is divided into regions which is further divided into groups. Each group has individuals in different states.



Simulation Scenario

- Regions
- Groups
- Individuals



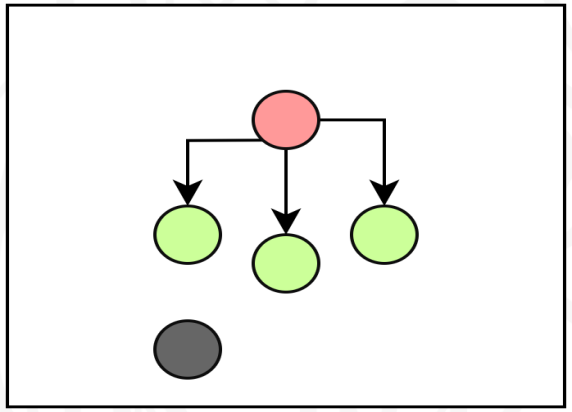
Simulation Steps

- Spread with in the group
- Spread across groups
- Caution against spread

Simulation Steps

1. **Spread with in the group:**

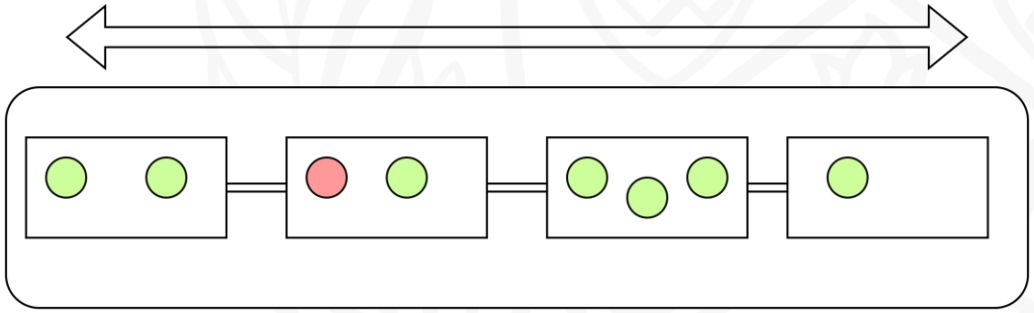
   In the first step, we simulate the scenario of an infected individuals spreading infection to susceptible individuals within the same group.



Simulation Steps

1. Spread with in the group
2. **Spread across groups :**

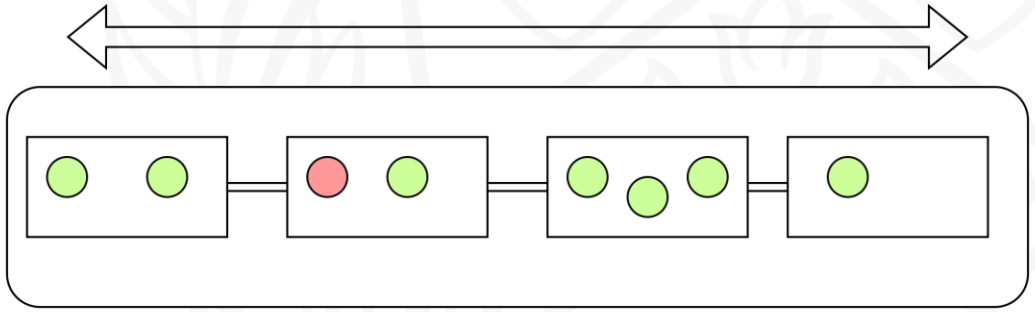
   In the second step, we simulate infected individuals moving across the groups and spreading infection in other groups.



Simulation Steps

1. Spread with in the group
2. Spread across groups
3. **Caution against spread:**

   In the third step, each region checks the number of infected individuals among its groups and if the value exceeds a certain threshold safety measures are enforced.



Parallelization

1. **Spread with in the group:**

   This step can be parallelized without any communication across processors as each processor has group information needed to simulate the step 1, so this step can be done simultaneously and in parallel across all processors.

Parallelization

1. **Spread with in the group**
2. **Spread across groups:**

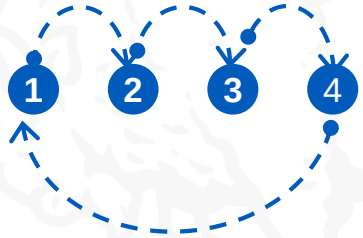
   If the infected individual travels across groups that are handled by different processors, then processors communicate to simulate this step. If both the groups are within same processor memory, then no communication is required.

Parallelization

1. **Spread with in the group**
2. **Spread across groups**
3. **Caution against spread:**

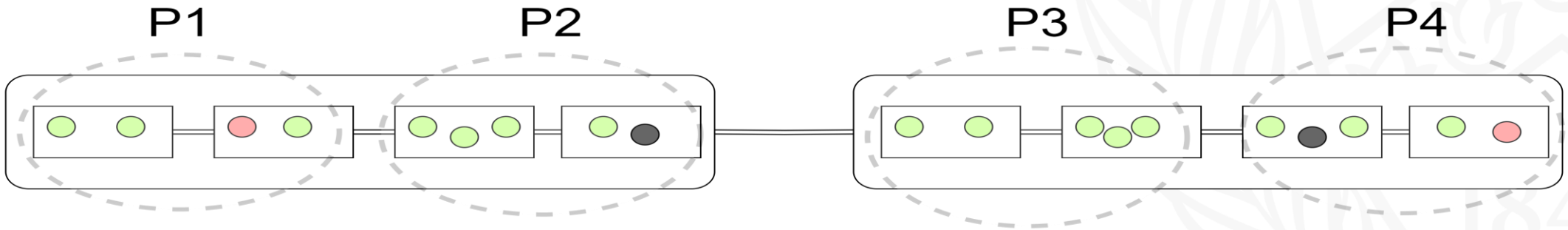
   case 1: If a processor has all the region information in its memory(as shown in figure below), then it can calculate the total infected individuals in a region by itself without communicating with other processors.

Parallelization

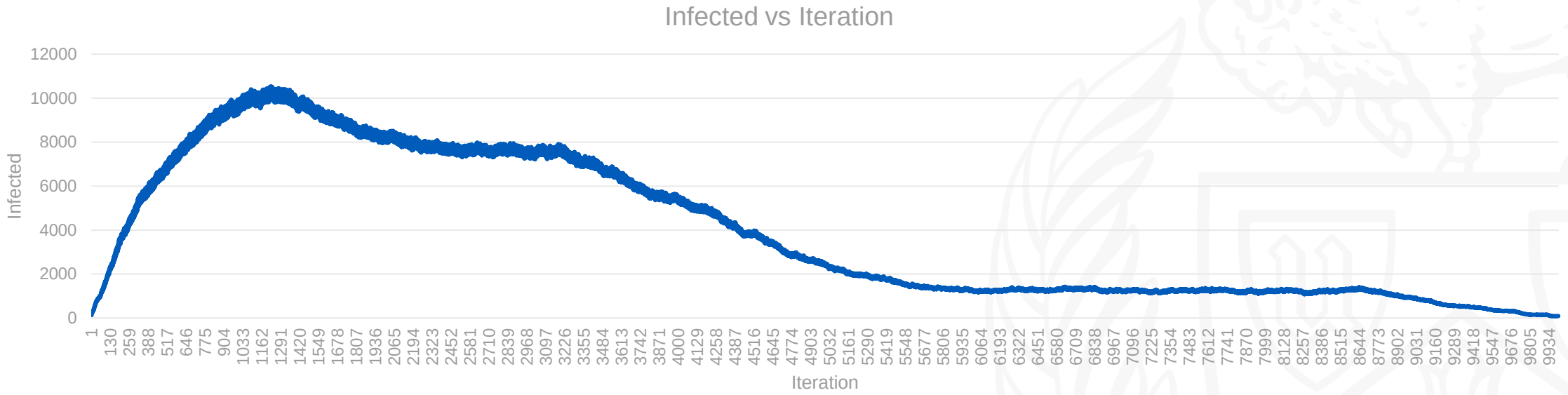


1. **Spread with in the group**
2. **Spread across groups**
3. **Caution against spread:**

   case 2: If data of a region is spread across multiple processors as shown in figure below(e.g., region 1 is spread across P1 and P2) , then processors rotate the data across so that every processors sees the data of other processors corresponding to that region(e.g., P1 and P2 rotate data with each other) and finally each processor knows the total infected individuals in a region.



Observations: Simulation Result

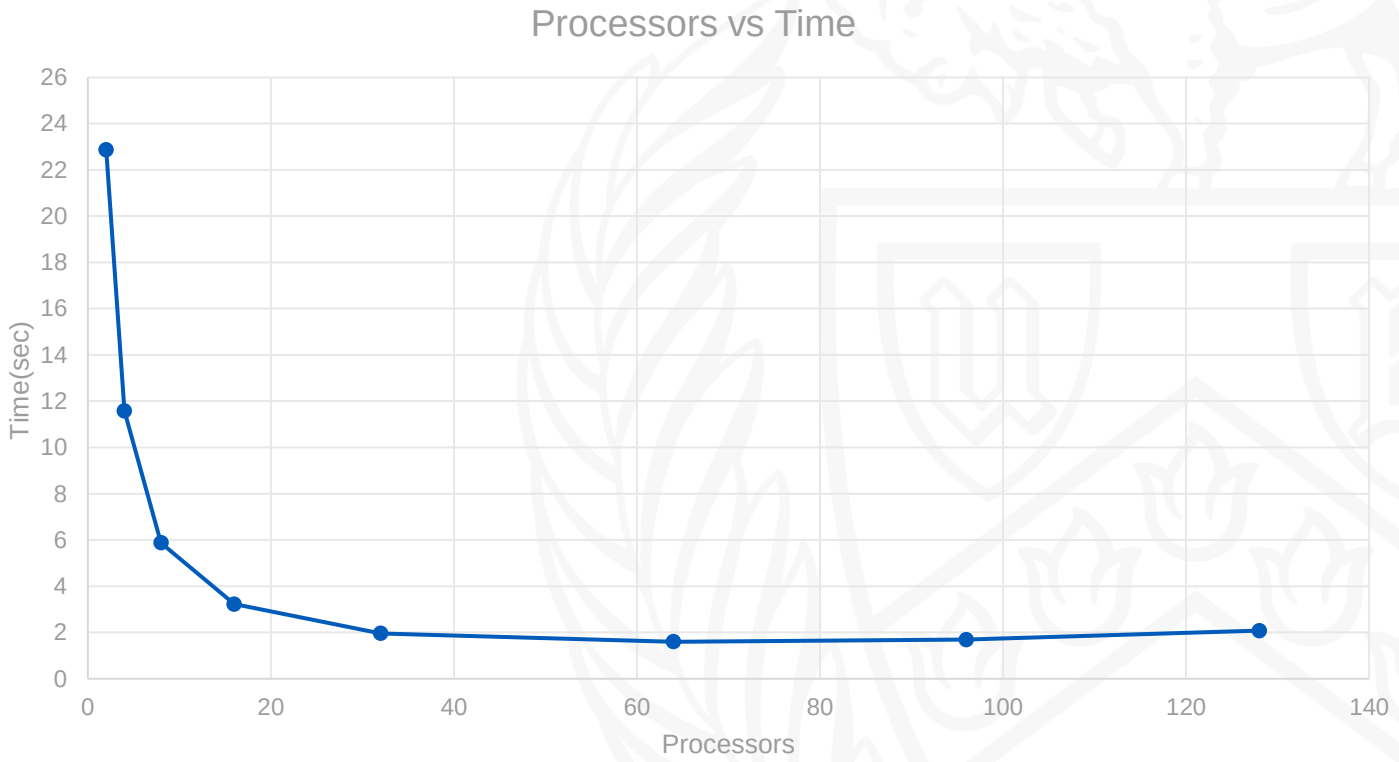


The result of disease spread simulation. Initially some individuals are affected among some groups. At the start the spread is high as there are lot of susceptible individuals. Then the infected count reaches the peak and regions try to enforce safety measure among groups and reduce the infection spread. After some iterations, some individuals start recovering and the number of susceptible individuals start decreasing so the number of new infected individuals decrease as well.

Processors vs Time Taken

1 Processor per Node 24k groups 2 regions

PE	Time(sec)	Data/PE
2	22.860463	12000
4	11.577422	6000
8	5.876857	3000
16	3.226291	1500
32	1.961257	750
64	1.595922	375
96	1.688871	250
128	2.078193	187



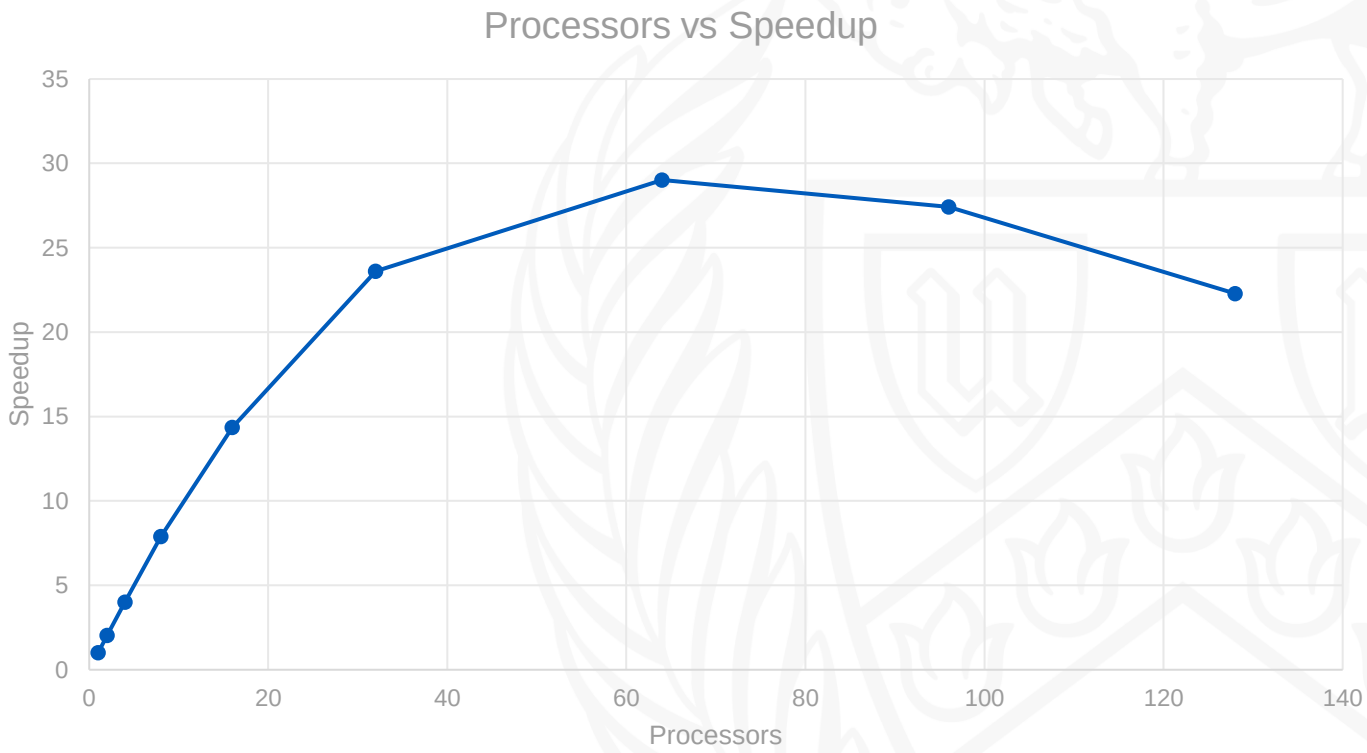
From 96 PE we can observe communication overhead starting to be more than processing time.

Speedup

1 Processor per Node 24k groups 2 regions

speedup = time(seq)/ time(parallel)

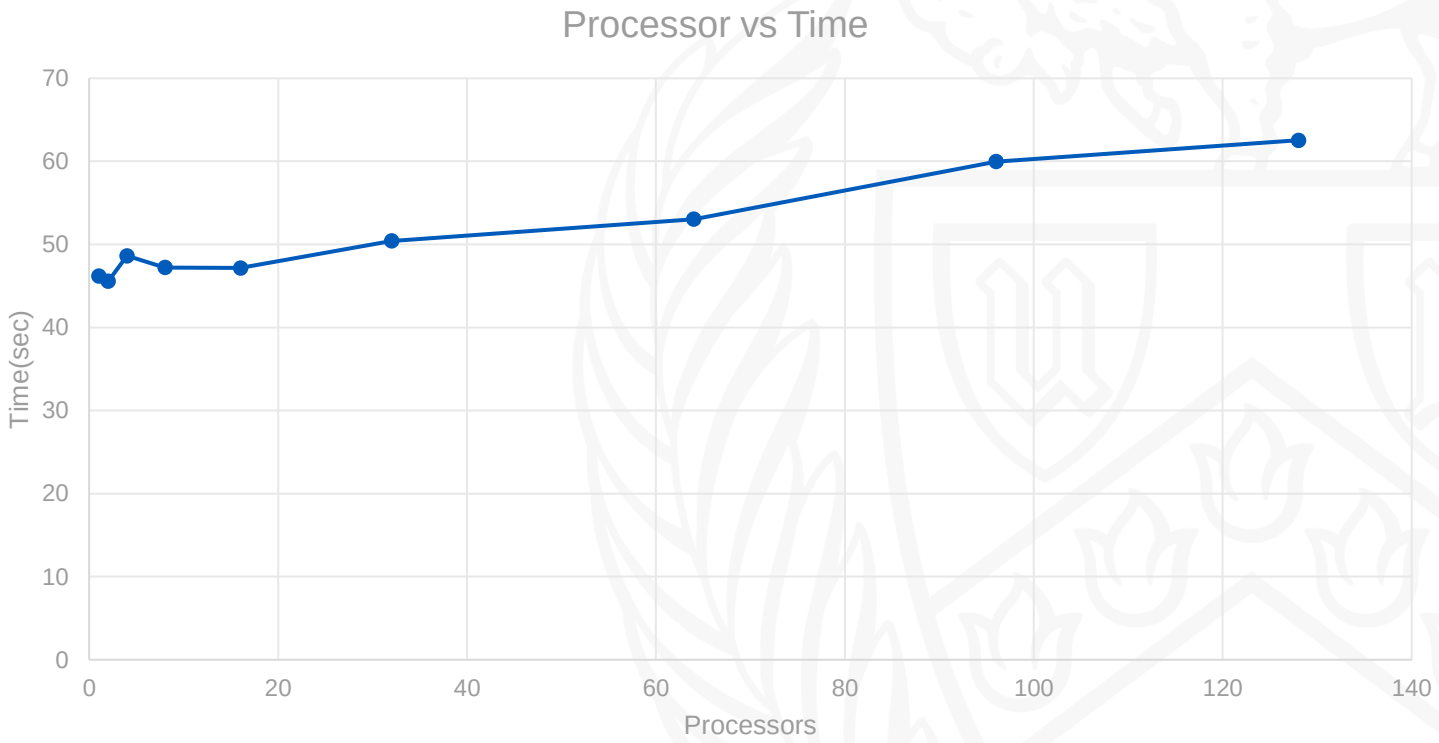
PE	Speedup	Data/PE
1	1	24000
2	2.02506594	12000
4	3.998640198	6000
8	7.877330519	3000
16	14.34896759	1500
32	23.60422168	750
64	29.00764887	375
96	27.41117883	250
128	22.27605665	187



Scaling (Gustafson's law)

1 Processor per Node

PE	Time(sec)	Data/PE
1	46.184201	24000
2	45.551088	24000
4	48.625592	24000
8	47.223639	24000
16	47.163421	24000
32	50.403299	24000
64	53.033097	24000
96	59.975084	24000
128	62.520124	24000

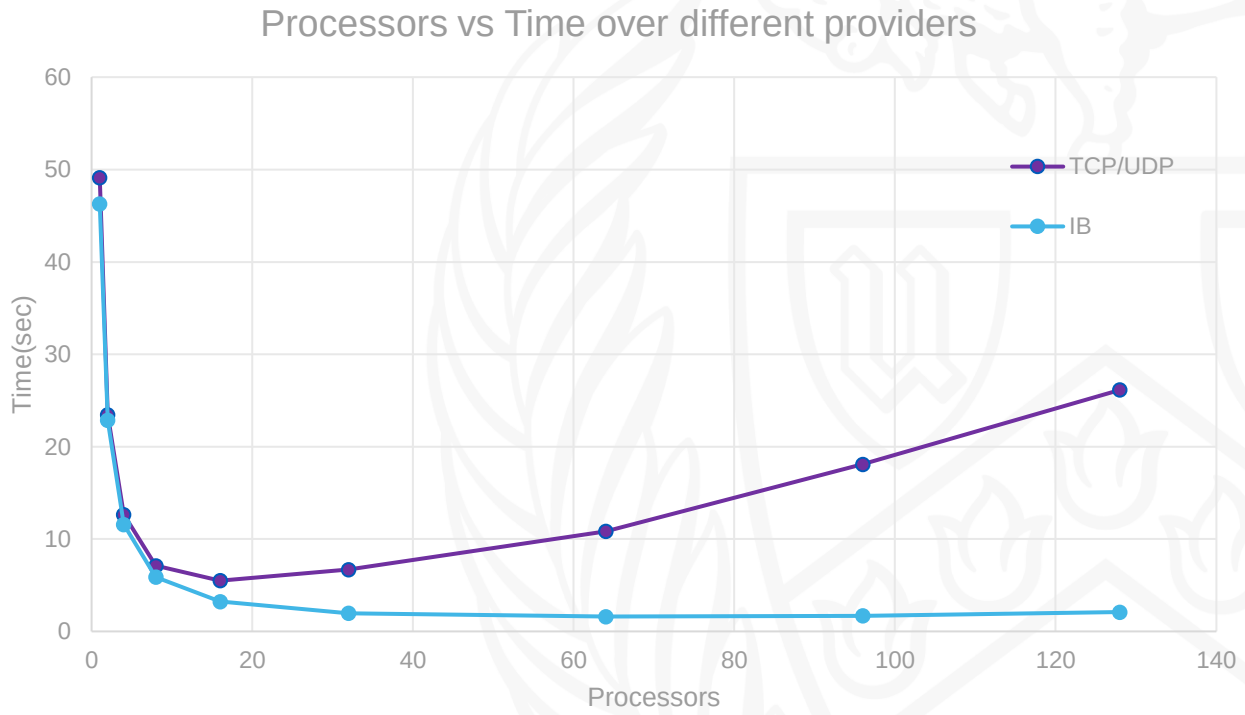


Total data for each run = 24000 * (No. of PE)

Processors vs Time Taken (IB vs TCP)

1 Processor per Node 24k groups 2 regions TCP/IP sockets vs IB

Processors	Time(sec) TCP/UDP	Time(sec) IB
1	49.1375	46.293945
2	23.456414	22.860463
4	12.652123	11.577422
8	7.102724	5.876857
16	5.504836	3.226291
32	6.697465	1.961257
64	10.826253	1.595922
96	18.1	1.688871
128	26.158287	2.078193



Comparing total time taken for the simulation to run using InfiniBand(IB) vs TCP/IP for communication.

Future Works

- Using OpenMP for parallelization
- Adding more extensions to the model
- Focusing more on how each factor affects the total spread

References

- https://ubccr.freshdesk.com/support/solutions/articles/13000026245-tutorials-workshops-and-training-documents
- Algorithms Sequential and Parallel: A Unified Approach (Dr. Russ Miller, Dr. Laurence Boxer).
- https://docs.ccr.buffalo.edu/en/latest
- How do mathematicians model infectious disease outbreaks?
- Mathematical modelling of infectious diseases