

Segundo proyecto programado

Simulación de distribución de CODE-VID

Eddy Ramírez Jiménez
Sistemas operativos - Universidad Nacional
2020

1. Motivación

Uno de los temas que se encuentran en este período de pandemia, como una necesidad real donde el trabajo de computación se ha vuelto mucho más especializado, es el análisis y creación de herramientas para atender la emergencia provocada por el famoso Coronavirus (SARS-CoV-2), responsable de generar la enfermedad llamada COVID-19 (Coronavirus disease 2019), que ha puesto en jaque a muchos países, jefes de estado y ha tenido un impacto muy serio en nuestras vidas y en la economía en general.

En este momento, se tienen computadoras que han hecho modelos tridimensionales de la estructura biológica del virus, en menos de una semana se contaba con el ADN (o más bien ARN) del virus (dato curioso, porque hace 100 años, en la última pandemia, ni siquiera se sabía qué cosa era el ADN), además, sabemos qué enzimas y qué proteínas encajan mejor con el virus y por ende, son aquellas células cuyo modelo epigenético (forma en como se ^{enrolla} el ADN y por lo tanto se manifiesta para transformar las proteínas ingeridas en proteínas humanas) permiten que el virus penetre en el cuerpo y empiece a deteriorarlo, donde particularmente las células del pulmón son muy receptivas a este tipo particular de coronavirus.

Otro de los modelos que ha servido es la generación de mapas de distribución del virus, donde se puede mostrar como se propaga una plaga, en este momento tenemos la suerte de que aunque el virus es altamente contagioso, es poco letal en comparación con otras pandemias que ha sufrido la humanidad, por ejemplo, la llamada Gripe Española, que contagió a un tercio de la población mundial y tenía una tasa de mortalidad del 20 %. O la que sería la peor plaga de la historia conocida, la *peste negra* que contagió a un 60 % de la población europea y tenía una tasa de mortalidad del 80 %.

Lo que sabemos es que hay muchos virus ahí afuera para los cuales la raza humana no ha tenido una inmunización oportuna, tenemos el problema de que por primera vez en la historia, por el fenómeno de la globalización, un virus puede llegar a todo el mundo (literalmente) en menos de 48 horas, que puede ser menos del tiempo de incubación que una enfermedad verdaderamente mortal pueda necesitar para manifestarse.

Ante esta necesidad generar modelos que simulen la forma en como se distribuye una enfermedad resulta sumamente útil, sobretodo si se pone en relación las características de diferentes poblaciones (por ejemplo, que una raza sea más propensa a contagiarse o a contagiar que otra).

Este proyecto busca precisamente hacer un modelo de simulación que provea de un esquema de como se puede difundir una enfermedad con una simplificación de las características básicas de diferentes subpoblaciones, con comportamientos diferentes.

2. Objetivos Formativos

Los objetivos de este proyecto se pueden encontrar en la carta al estudiante, sobretodo en el tema 2.

3. Especificación del proyecto

Su proyecto consiste en desarrollar un modelo sobre un mapa que muestre como ciertos agentes con comportamiento diferente (pero que se mueven con cierto patrón), se pueden ir contagiando de una enfermedad, a esta la vamos a llamar Code-vid.

3.1. Características de los agentes

En un plano como un malla, se van a colocar diferentes tipos de agentes, los cuales van a tener diferentes comportamientos y a ser moverse de acuerdo a su comportamiento particular. A continuación se detallan las propiedades de los agentes:

3.1.1. Tipos de agentes

En un mapa 2D se modelarán 4 tipos de agentes:

1. Rectos: viajan en línea recta, hasta chocar con pared y rebotar de forma simétrica con respecto a la normal que llevaban. (Personas que viajan sin punto definido)
2. Estacionales: Viajan siguiendo rutas prefijadas y luego, regresando al punto de origen (Equivalentes a rutas de autobuses)
3. Aleatorios: Se mueven aleatoriamente hacia puntos relativamente cercanos en el mapa. (Equivalentes a servicios de transporte privado)
4. Estáticos: No se mueven de donde se encuentran. (Equivalente a quienes guardan una cuarentena).

3.1.2. Estado en que se encuentra

Habrán tres tipos de posibles estados, primero sano, luego enfermo y finalmente curado. Si no sobreviviese, entonces, debe desaparecer del mapa. Hay algunos casos en que

3.1.3. Probabilidad de infectarse

Todos estos agentes, van a tener una probabilidad de contraer el virus si se cruzaran en el mismo punto (x, y) con alguien que tenga el virus.

3.1.4. Condición inicial

Van a haber dos posibles condiciones iniciales, sano o enfermo.

3.1.5. Probabilidad de morir o curarse

En cada simulación, luego del tiempo de enfermedad, puede que se cure o puede que muera. Cada hecho, tendría tiempos independientes (es decir, hay un tiempo en que se está enfermo y se sobrevive y otro tiempo en que se está enfermo y se muere).

3.2. Características del mapa

El mapa debe de ser una malla, donde se mueven los agentes, los cuales deben dispersarse aleatoriamente y en principio, no habría problema con que dos agentes coincidan en una casilla particular.

El mapa debe ser matricial y por ende tener un largo y ancho definidos. Puede que en el mapa hayan paredes u obstáculos que agentes no pueden cruzar (simulando políticas de restricción o confinamiento).

A continuación se describe lo que ocurre cuando un agente se encuentra con una pared en su ruta, dependiendo de su tipo:

1. Rectos: se comportan tal cual se encontraran con el fin de la matriz, rebotando de forma simétrica a la normal de la pared.
2. Estacionales: Se regresan en su ruta, dejando la otra parte de su ruta sin completar.
3. Aleatorios: Vuelven a buscar un punto aleatorio hasta que éste no implique estar volver a toparse con un muro.
4. Estáticos: Éstos no deberían de verse afectados por una pared en el mapa.

3.3. Entrada del programa

El programa va a tener 3 archivos de configuración diferentes. Uno para agentes, otro para el mapa y otro para las características propias del codevid.

3.3.1. Configuración de agentes

El programa va a generar cierta cantidad de agentes según su tipo en diferentes posiciones aleatorias. El archivo en su primera línea tendrá la cantidad de grupos de agentes se van a realizar.

La primer línea de cada grupo tiene la cantidad de agentes con esas características. La segunda línea tendrá el tipo de agente (1- Rectos, 2- Estacionales, 3- Aleatorios y 4 Estáticos). La siguiente línea indicando su velocidad máxima y mínima. La última línea será una *s* de sano o una *e* de enfermo, indicando si ese grupo está sano o enfermo.

Todos los agentes deben ser hilos que según su tipo, harán su trayectoria como mejor les convenga en el mapa, según su tipo. Los de tipo Estacionales, deben tener al menos 7 puntos que visitar, que serán generados cuando el hilo se genera. Los de tipo rectos, deben tener un vector que les marque el rumbo (en (x, y)).

Ejemplo:

5
3
1

```

10 30
e
6
2
15 20
s
12
3
5 20
s
40
4
0 0
s
20
3
4 10
s

```

3.3.2. Configuración de mapa

El mapa debe tener en su primer línea el largo y ancho de la matriz por la que se moverán los agentes. La segunda línea tendrá la cantidad de paredes que se deberán colocar P y las siguientes P líneas tendrán cuatro números x_1, y_1, x_2, y_2 indicando las coordenadas superior izquierda e inferior derecha del rectángulo que sería esa pared. Nótese que todas las paredes son rectangulares.

Ejemplo del mapa:

```

700 500
2
100 1 101 200
100 250 101 500

```

3.3.3. Configuración de Code-vid

Finalmente el archivo de configuración del CodeVid, indicará algunos datos importantes del comportamiento de la enfermedad. La primer línea indicará la probabilidad (flotante) de que un individuo muera, la segunda

línea el tiempo en segundos que tarda una persona en morir, en la tercera línea el tiempo en segundos que tarda una persona en sanar. Las siguientes cuatro líneas tendrán la tasa de contagio en forma de matriz para los diferentes tipos de agentes y como se contagia la enfermedad. La posición (i, j) indica la probabilidad de que un individuo de tipo i contagie a alguien de tipo j . Finalmente la última línea, indicará si es posible una reinfección, cero para indicar que no, una persona curada, no se puede enfermar de Code-vid de nuevo y cualquier otro número para que esto sí pueda ocurrir.

Ejemplo de datos del Code-vid:

```
3.45
18
20
99.5 99.5 99.5 99.5
99.5 99.5 99.5 99.5
99.5 99.5 99.5 99.5
99.5 99.5 99.5 3.5
0
```

3.4. Entrada de la ejecución del sistema

El programa al ejecutarse se debe escribir cuál es la duración del programa.

Ejemplo:

```
./progra 200
```

3.5. Salida Esperada

El programa debe de escribir un archivo de Látex donde se muestren gráficos, segundo a segundo de cómo está el mapa en cada momento, durante la duración de la simulación. Indicando con un punto verde a aquellos agentes sanos y con un punto rojo a aquellos agentes enfermos. Para ello puede utilizar cualquier paquete de Látex que considere conveniente.

Además, al finalizar los gráficos, debe de mostrar un diagrama donde el eje de las ordenadas son el tiempo (en segundos) y el eje de las abscisas, es el número de contagiados que hay en el tiempo.

Este archivo LáTeX debe tener una portada con los autores del proyecto programado y desde luego, debe ser compilable a un PDF.

4. Metodología

El proyecto consta de dos partes, por lo que la simulación debe darse con todos los agentes siendo hilos y teniendo particular cuidado con la región crítica que es la matriz y sobretodo con las probabilidades y las velocidades. (puede que una velocidad particular haga que aunque un agente se salte algunas casillas de su ruta).

Luego deben de investigar correctamente como hacer lo que se solicita en Latex.

5. Rúbrica

Producto esperado	Porcentaje
Simulación con todo lo especificado	75 %
Archivo Latex resultante	25 %
TOTAL	100 %

6. Estimación de tiempo

- El proyecto se hará en los mismos equipos de trabajo que se realizarán en la edición anterior.
- Fecha de entrega: 25 de mayo

7. Aspecto Generales

- Formato de entrega: Deben entregar en un correo a eddy.ramirez.jimenez@una.cr el o los archivo(s) con un makefile.
- Penalizaciones: Se penalizará con 5 puntos menos de la nota por cada hora de entrega tardía del proyecto.