ESCUELA DE CIENCIAS DE LA
COMPUTACIÓN E INFORMÁTICA
UNIVERSIDAD DE COSTA RICA

Proyecto con MPI / C++

Objetivos

Que el estudiante profundice en:

1. El diseño de programas paralelos “escalables” basados en memoria distribuida identificando y combinando los patrones de programación paralela estructurada adecuados tales como: fork, join, map y reduction.
2. La implementación de un programa paralelo identificando y combinando funciones y tipos de MPI en lenguaje C++.
3. La depuración sistemática de un programa paralelo hasta lograr el funcionamiento esperado utilizando algún depurador simbólico (asociado al IDE de su preferencia) o haciéndolo desde la consola mediante la interfaz de "gdb" por "comandos".
4. La evaluación del desempeño de un programa paralelo utilizando conceptos simples como el “tiempo pared”, “aceleración” y “eficiencia” para que sea capaz de mejorarlo.

Descripción del problema

Se deberá elaborar un programa paralelo basado en MPI/C++ que realice una simulación de un proceso de infección de un virus en un grupo de personas tal como se puede visualizar en: http://netlogoweb.org/launch#http://netlogoweb.org/assets/modelslib/Sample%20Models/Biology/Virus.nlogo.

Los parámetros de entrada de la simulación son:

1. Cantidad de personas (number-people): [0..10,000,000].
2. Probabilidad infecciosa de virus(infectiousness): [0..1].
3. Probabilidad de recuperación (chance-recover): [0..1].
4. Duración máxima de la infección antes de morir: [5..50].
5. Porcentaje de personas originalmente infectadas: [0..10].
6. Tamaño del espacio bidimensional: a) 100x100, b) 500x500, c) 1000x1000.

Los resultados o salida de la simulación son:

1. Promedio por tic, porcentaje y cantidad actual de personas infectadas.
2. Promedio por tic, porcentaje y cantidad actual de personas susceptibles.
3. Promedio por tic, porcentaje y cantidad actual de personas recuperadas.
4. Promedio por tic, porcentaje y cantidad actual de personas muertas.
5. Cantidad total de tics, “tiempo pared” y “tiempo pared por tic”.

Los supuestos son que:

1. Las personas sólo se pueden morir por la infección.
2. No pueden nacer nuevas personas.
3. Una vez recuperadas, las personas nunca más pueden infectarse.
4. La simulación termina cuando no hayan personas infectadas.

Las reglas que rigen el comportamiento de las personas son:

1. Deambulan al azar por el espacio bidimensional, trasladándose, por tic, a una posición en cualquier dirección a partir de su posición actual.
2. Una persona infectada sólo puede contagiar a otras que ocupen su misma posición en el mismo tic, de acuerdo con la probabilidad de infección y el estado de las personas con que comparte la posición. Si coincidieran dos o más personas infectadas, junto con otras, en una posición, la probabilidad de infección se incrementa linealmente de acuerdo con la cantidad de personas infectadas que coinciden.
3. Una persona puede morir sólo estando infectada cuando se cumple la duración máxima de la infección.

De acuerdo con lo anterior usted deberá elaborar un programa modularizado por funciones en el que las responsabilidades del proceso cero sean:

1. Capturar y validar los valores de los parámetros de entrada.
2. Generar mensajes de error para el usuario si los valores de los parámetros no son correctos.
3. Generar un archivo con las salidas por tic o día y mostrarlas por consola.
4. Simular el proceso de infección.
5. Opcionalmente (para obtener puntos extra) podrá generar una salida gráfica que muestre en colores la curva verde para la variable “cantidad de susceptibles en el tic t”, gris para la curva de la variable “cantidad de resistentes en el tic t” y rojo para la variable “cantidad de infectados en el tic t”.

Los demás procesos deberán concentrarse en las funciones de la simulación.

Criterios de evaluación

La calidad de su programa se valorará con base en los siguientes criterios:

1. **Escalabilidad demostrada**: su programa será ejecutado en nuestro “cluster” arenal con 2, 4 y 8 procesos utilizando 1, 2 y 4 nodos respectivamente.
2. **Desempeño demostrado**: su trabajo será comparado con aquél que muestre el **mejor desempeño por tic** tomando en cuenta al menos dos diseños de experimento. Entiéndase por “diseño de experimento” un conjunto de valores de parámetros. Ver tabla de análisis de desempeño al final de este documento.
3. **Eficacia demostrada**: su trabajo será comparado en las salidas que genera con la versión web referenciada, para lo cual se utilizará al menos dos diseños de experimentos.
4. Eficiencia en el uso de memoria, basado en las estructuras de datos utilizadas.
5. Simplicidad del código.
6. Forma y estilo del código: sangrado de bloques de código, nombres de objetos empiezan en minúsculas, nombres de clases empiezan en mayúsculas, nombres de métodos también empiezan en minúscula pero se usan mayúsculas para concatenar palabras, comentarios para los atributos y variables de métodos.
7. División de responsabilidades entre el proceso maestro y los subordinados.

Fecha de entrega: domingo 2 de diciembre a las 23:55 por medio del enlace en el sitio del curso. SÓLO DEBERÁ SUBIR LOS ARCHIVOS DE CÓDIGO FUENTE (*.h y *.cpp) los archivos de salida de al menos tres ejecuciones.

Evaluación

Criterio	%Prc
Escalabilidad demostrada	20
Desempeño demostrado	40
Eficacia demostrada	40
Hasta 10/100 puntos extra por tarea graficadora	10
Hasta 10/100 puntos extra por un buen reporte de errores cuando NO funcione	10

Notas importantes:

1. Si el programa NO funciona, tendrá cero puntos en todos los rubros porque no será posible demostrar ninguna de las características.

2. Este proyecto deberá realizarse idealmente y a lo más en parejas. NO SE ACEPTARÁ NINGÚN TRABAJO ELABORADO POR MÁS DE DOS PERSONAS.

3. Cada hora de atraso en la entrega se penalizará con -1/100, lo que se aplicará a la nota obtenida.

4. A TODOS LOS ESTUDIANTES INVOLUCRADOS EN UN FRAUDE SE LES APLICARÁ EL ARTÍCULO #5 INCISO C DEL "Reglamento de Orden y Disciplina de los Estudiantes de la Universidad de Costa Rica".

5. NO SUBA ningún otro archivo que no sea de código fuente (*,h y *.cpp) o de datos para evitar la transmisión de virus.

Tabla de análisis de desempeño, aceleración y eficiencia: en las filas la cantidad de procesos, en las columnas la cantidad de nodos. En celeste MPI_Wtime() ejecutado en nuestro "cluster" “arenal”, en amarillo la aceleración respecto de la versión 2x1 y en verde la eficiencia respecto de la versión 2x1. Cada “tiempo pared” representa el promedio por tic que ha durado la simulación, o sea, “tiempo pared” total dividido por cantidad de tics que duró la simulación.

PxN	2x1	4x2	8x4	2x1	4x2	8x4	2x1	4x2	8x4
exp1									
exp2									

	exp#1	exp#2
Población	1,000,000	10,000,000
Dimensión	500x500	1000x1000
Infección inicial	10%	10%
Duración	20	20
Probabilidad infección	0.65	0.65
Probabilidad recuperación	0.50	0.50