

## Proyecto con OpenMP / C++

### Objetivos

Que el estudiante profundice su aprendizaje sobre:

1. El diseño de programas paralelos, eficientes y “escalables” basados en hilos y memoria compartida usando OpenMP y C++.
2. La implementación de un programa que mezcla paralelización por datos y tareas.
3. La depuración sistemática de un programa paralelo.
4. La evaluación del desempeño de un programa paralelo utilizando las métricas: “tiempo pared”, “aceleración” y “eficiencia” para mejorarlo.

### Descripción del problema

Se deberá elaborar un programa paralelo multitarea usando OpenMP y C++ que realice una simulación de autómatas celulares como en: <https://www.youtube.com/watch?v=IK7nBOLYzdE>. Esta simulación se basa en un autómata de una dimensión con la regla 30 (a1d) y el autómata de dos dimensiones de Conway (a2d).

Las **entradas por línea de consola** del programa deben ser:

1. la cantidad de hilos por núcleo: **chxn**,
2. la cantidad de bits de ambos autómatas: **N** (se supone que la matriz para el autómata de Conway es cuadrada, NxN),
3. la cantidad de iteraciones **I**.

La **salidas** del programa serán:

1. un archivo de texto con las líneas de bits que genera a1d,
2. un archivo de texto con las matrices de bits que genera a2d,
3. los valores de las variables de “tiempo pared” para generar la “Tabla de desempeño” que se describe en la sección de evaluación.

### Restricciones:

1. el programa deberá basarse en **al menos dos** tareas independientes, una para cada autómata celular,
2. las tareas deberán interactuar mediante **al menos una** cola, la fila provista por a1d alimenta al a2d en la fila cero,
3. **en cada iteración** a2d alimenta la fila cero con una línea provista por a1d, a menos que la cola esté vacía,
4. se deberán crear tantos hilos como:  $chxn * \text{cantidad de núcleos}$ ,
5. los hilos se dividirán equitativamente entre las tareas para obtener el mejor “tiempo pared” y escalabilidad.

El programa provisto por el docente y basado en mpi4py puede servir para corroborar los resultados del programa desarrollado y como esquema para la solución.

Las **reglas** de los autómatas son:

1. regla 30 del ald, descrita en <https://mathworld.wolfram.com/Rule30.html>,
2. la del autómata de Conway, descrita en [https://es.wikipedia.org/wiki/Juego\\_de\\_la\\_vida](https://es.wikipedia.org/wiki/Juego_de_la_vida).

Para una explicación completa de qué son los autómatas celulares ver: [https://es.wikipedia.org/wiki/Aut%C3%B3mata\\_celular](https://es.wikipedia.org/wiki/Aut%C3%B3mata_celular).

### **Criterios de evaluación**

Su programa será evaluado con base en los siguientes cinco criterios básicos, simplicidad, modularidad, eficacia, eficiencia y escalabilidad:

- Se entiende por “simplicidad” que el programa sea lo más fácil de entender, de depurar y de modificar que sea posible.
- Se entiende por “modularidad” la correcta separación de funciones entre el código de los distintos “módulos”. Un módulo es una clase, una función o grupo de funciones, y la función principal “main()”. Por ejemplo, con una adecuada modularidad el “main()” se encarga de la entrada y salida de datos, así como la generación de mensajes por consola dirigidos al usuario. Las clases NO deben encargarse de las funciones del “main”.
- Se entiende por “eficacia” que el programa cumpla con el objetivo de simular el proceso infeccioso correctamente con base en las reglas anteriores.
- Se entiende por “eficiencia” que el programa cumpla con el objetivo en el menor “tiempo pared” posible.
- Se entiende por “escalabilidad” que:
  - el código se adapte a la cantidad de núcleos y a la cantidad de hilos indicada por el usuario por línea de consola,
  - la “eficiencia” mejora cuando se agregan más núcleos, y más hilos, aunque en algún momento se cumpla la ley de Amdahl.

La evaluación de la eficiencia y escalabilidad será **comparativa**, lo que significa que su puntaje se establecerá en comparación con los demás trabajos presentados en el grupo:

1. La eficiencia se medirá en términos de qué tan rápido es su programa con base en la comparación del “tiempo pared” con los demás trabajos de su grupo.
2. La escalabilidad se medirá en términos de si su programa mejora significativamente (aceleración o “speedup”) cuando se incrementa la cantidad de hilos que variará en {8, 16}.

Dado que hay muchos factores que intervienen en cada ejecución de un programa, la única forma de comparar las mediciones o métricas especificadas de un programa con las de otros es mediante **promedios simples**: “promedio de eficacia”, “promedio de tiempo pared” y “promedio de aceleración”. Por tanto usted deberá adjuntar a la entrega de su trabajo un archivo pdf con la siguiente tabla a efecto de que su programa pueda ser debidamente comparado con los demás.

**Tabla de desempeño**

8 hilos (ejm: 4núcleos x 2hilos)		16 hilos (ejm: 4núcleos x 4hilos)			
$I = 10^6, N = 10^3$	$I = 10^8, N = 10^5$	$I = 10^6, N = 10^3$		$I = 10^8, N = 10^5$	
$tp_{11}$	$tp_{12}$	$tp_{21}$	$ac_1$	$tp_{22}$	$ac_2$
de- $tp_{11}$	de- $tp_{12}$	de- $tp_{21}$	de- $ac_1$	de- $tp_{22}$	de- $ac_2$

Donde (todos son promedios **en segundos**):

- $tp_{1i}$  == es el promedio simple de los “tiempo pared” en diez ejecuciones con 8 hilos, se deberá usar segundos como unidad de tiempo,
- $tp_{2i}$  == es el promedio simple de los “tiempo pared” en diez ejecuciones con 16 hilos,
- $ac_i$  ==  $tp_{1i} / tp_{2i}$ , que representa el promedio de las aceleraciones en diez ejecuciones al pasar de 8 a 16 hilos.
- La última fila representa las desviaciones estándar de  $tp$  y  $ac$ .

El asistente a cargo validará la tabla aportada y asignará puntaje<sup>1</sup> a cada trabajo, en los rubros de eficiencia y escalabilidad, ordenando la lista de valores correspondiente de menor a mayor y luego dividiéndola en mínimo dos (si la desv-st es pequeña) y máximo cuatro grupos (si la desv-st es grande) con resultados similares, luego asignará:

9 o 10: para los mejores promedios,  
7 u 8: para los promedios buenos,  
6: para los promedios regulares, y  
menos de 6: para los promedios malos.

**Tabla de Evaluación (se entrega 30 de noviembre a las 23:59 vía Mediación Virtual)**

Criterio	%
Simplicidad	5
Modularidad	5
Eficacia	30
Eficiencia	35
Escalabilidad	25
Hasta 10 puntos por un reporte de errores completo <b>en caso que no haya funcionado</b> el programa	

**Notas importantes:**

1. Si el programa no funciona, es decir la simulación es incorrecta, no obtendrá más de 5/100 puntos, o 10/100 si incluye un reporte de errores completo. Obtendrá cero puntos en todos los rubros.
2. Si no aparece la tabla de desempeño el trabajo no podrá obtener más de 5/35 en los rubros de eficiencia y escalabilidad.
3. Este proyecto deberá realizarse idealmente y a lo más en parejas. NO SE ACEPTARÁ NINGÚN TRABAJO ELABORADO POR MÁS DE DOS PERSONAS.
4. Cada hora de atraso en la entrega se penalizará con -1/100, lo que se aplicará a la nota obtenida.
5. 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".
6. NO SUBA ningún otro archivo que no sea de código fuente (\*.h, \*.cpp) o de datos y resultados para evitar la transmisión de virus.

<sup>1</sup> Para validar los datos el asistente realizará al menos una prueba con  $I=100$  y  $N=10$ .