
SIMULACIÓN USANDO AUTÓMATAS CELULARES CON MULTIPROCESAMIENTO

A PREPRINT

Rios Gastón Gustavo
Facultad de Informática
Universidad Nacional de La Plata
okason1997@hotmail.com

March 15, 2019

ABSTRACT

Se realizó la simulación de las células Dictyostelea utilizando autómatas celulares e implementando multiprocesamiento con memoria compartida. También se realizó el análisis y comparación de una posible solución implementando pasaje de mensajes.

1 Introducción

Un autómata celular es un sistema computacional discreto compuesto por un número finito de unidades simples homogéneas llamadas células o átomos. En cada unidad de tiempo, las células poseen un estado, el cual cambiará en todas en paralelo en cada paso mediante una función de actualización, la cual tendrá en cuenta el estado de las celdas vecinas. Este tipo de comportamiento favorece en gran medida la paralelización de sus células. Los autómatas celulares tienen múltiples usos posibles, entre ellos la simulación de sistemas dinámicos. En mi caso elegí simular las células llamadas Dictyostelea. Estas son un grupo de protistas conocidos comúnmente como hongos mucilaginosos. Estos organismos pueden tomar tres etapas en su vida: una etapa unicelular, una etapa de agregación y una etapa de cuerpo fructífero formador de esporas. Esto le permite pasar de ser un organismo unicelular a multicelular. Este comportamiento lo realiza mediante la interacción con sustancias químicas del entorno midiendo su concentración para decidir qué camino tomar. Este comportamiento será la función de actualización de las células del autómata celular, donde cada célula, midiendo la concentración de sustancias químicas a su alrededor decidirá dónde moverse.

2 Memoria compartida vs Pasaje de mensajes

Queda clara la posibilidad de paralelizar los autómatas celulares, ya que se poseen múltiples células realizando la misma operación en paralelo en una unidad de tiempo, por lo que paralelizarlas otorgaría idealmente un gran speedup. La complejidad se encuentra en que las células deberán comparar todas contra la misma matriz, la cual será modificada en cada paso. También hay que tener en cuenta la colisión de células, al moverse en paralelo y siguiendo la concentración de sustancia es muy frecuente la colisión de células, lo que podría causar la pérdida de alguna célula de no ser tratado el caso. En mi caso decidí que en caso de colisión una de las células no se mueva o busque modificar su camino para moverse hacia el químico.

2.1 Pasaje de mensajes

El pasaje de mensajes permite la sincronización de las células enviando mensajes entre los procesos del programa de forma sincrónica (bloqueante) o asincrónica (no bloqueante). Posibles soluciones utilizando pasaje de mensajes serían la división de la matriz en n partes a ser procesadas por n procesos usando pasaje de mensaje en las zonas de conflicto de los procesos para sincronizar. El problema de esta solución sería la carga despareja ocasionada por la diferencia en la cantidad de células Dictyostelea de cada división. Otra posible solución con pasajes de mensajes sería la utilización del patrón maestro esclavo, donde el maestro les otorgaría a cada proceso esclavo una cantidad de células Dictyostelea n , y

cada proceso esclavo realizará las funciones de actualización necesarias devolviendo los resultados al maestro, quien resolverá las colisiones. El problema de ambas soluciones de pasaje de mensajes es que al aumentar el tamaño de la matriz, el número de zonas de conflicto aumentan en gran medida, lo que generaría una gran cantidad de mensajes pasados entre los procesos, o una sobrecarga del proceso maestro en la segunda solución.

2.1.1 Memoria compartida

Por las razones especificadas previamente es más conveniente la utilización de memoria compartida. La memoria compartida implica que todos los procesos comparten una misma memoria y tienen acceso a esta, lo que permite la utilización de una matriz compartida en la cual las células podrían indicar donde harán su siguiente movimiento. El acceso a esta matriz será bloqueante, para evitar conflictos de movimiento. La posibilidad de tener memoria compartida también permite poseer matrices más grandes sin sobrecargar tanto la memoria, ya que si no fuera compartida se debería tener una matriz por cada proceso o un pasaje de mensaje cada vez que se quisiera obtener datos de la matriz, lo que generaría nuevamente una gran cantidad de mensajes en circulación y generaría un cuello de botella en el maestro.

3 Implementación

Para la implementación del programa se utilizó el lenguaje cython y matplotlib para dibujar la matriz. El programa consiste de células ubicadas en un mundo, el cual es una matriz. En este mundo también hay comida, la cual estará rodeada de sustancias químicas que atraen a las células a esta. Las células se moverán por el mundo buscando toda la comida posible para alimentarse. A medida que se mueven, las células irán gastando energía y cuando esta se agote emitirán una señal química a las demás células indicando una concentración baja de comida, las células que reciban esta señal comenzarán la etapa de agregación, en la cual emitirán la misma señal química que recibieron y se moverán hacia la mayor concentración de este químico para formar la etapa de cuerpo fructífero formador de esporas. Se utilizan 2 matrices, `world` y `entity_map`: `world` poseerá la concentración de sustancias químicas del mundo, las células utilizarán estas concentraciones para decidir hacia donde moverse, y `entity_map` posee la matriz donde están ubicadas las células y la comida. Al comenzar el programa se inicializarán las células y la comida en sitios al azar del mundo. Por cada comida situada se situará en `world` su señal química correspondiente. En cada instancia de tiempo se correrá en paralelo el procesamiento de cada célula Dictyostelea, las cuales han sido especificadas en un array para mejorar el paralelismo dando a cada proceso la misma cantidad de células Dictyostelea. La sincronización entre las células está dada por locks para las celdas de las matrices `world` y `entity_map`, de forma que múltiples células Dictyostelea pueden trabajar sobre estas matrices en paralelo sincronizando el acceso. El procesamiento de cada célula Dictyostelea consta de los siguientes pasos: Exploración: se mirará cada valor vecino en el `world` para descubrir algún rastro químico el cual seguir. Movimiento: en caso de haber descubierto algún rastro químico, se irá hacia el rastro más fuerte. Si se genera una colisión se intentará moverse a una celda adyacente a la seleccionada. Si se encuentra comida, se eliminará el rastro de esta y aumenta la energía de la célula Dictyostelea. En caso de no haber descubierto rastro, se moverá de forma al azar intentando encontrar un rastro. Agregación: si la célula no posee más energía o ha recibido una señal química de agregación de otra célula Dictyostelea, se generará una señal química de agregación.

3.0.1 Performance

Utilizando `profile` de python tome el tiempo de 100 iteraciones del programa obteniendo los siguientes resultados:

Parámetros	Resultados
Tamaño del mundo = 256*256 Cantidad de threads = 1 Cantidad de células Dictyostelea = 1024	Tiempo por iteración = 0.390 Tiempo total = 39.017
Tamaño del mundo = 256*256 Cantidad de threads = 2 Cantidad de células Dictyostelea = 1024	Tiempo por iteración = 0.202 Tiempo total = 20.154
Tamaño del mundo = 256*256 Cantidad de threads = 4 Cantidad de células Dictyostelea = 1024	Tiempo por iteración = 0.114 Tiempo total = 11.406
Tamaño del mundo = 256*256 Cantidad de threads = 6 Cantidad de células Dictyostelea = 1024	Tiempo por iteración = 0.083 Tiempo total = 8.263

References

- [1] Cellular Automata. <https://plato.stanford.edu/entries/cellular-automata/> 14/3/2019.
- [2] Cython. <https://cython.readthedocs.io/en/latest/> 14/3/2019.
- [3] Dictyostelea. <https://es.wikipedia.org/wiki/Dictyostelea> 14/3/2019.
- [4] A model for individual and collective cell movement in Dictyostelium discoideum. <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC27044/> 14/3/2019.