Proyecto con MPI4PY / Python

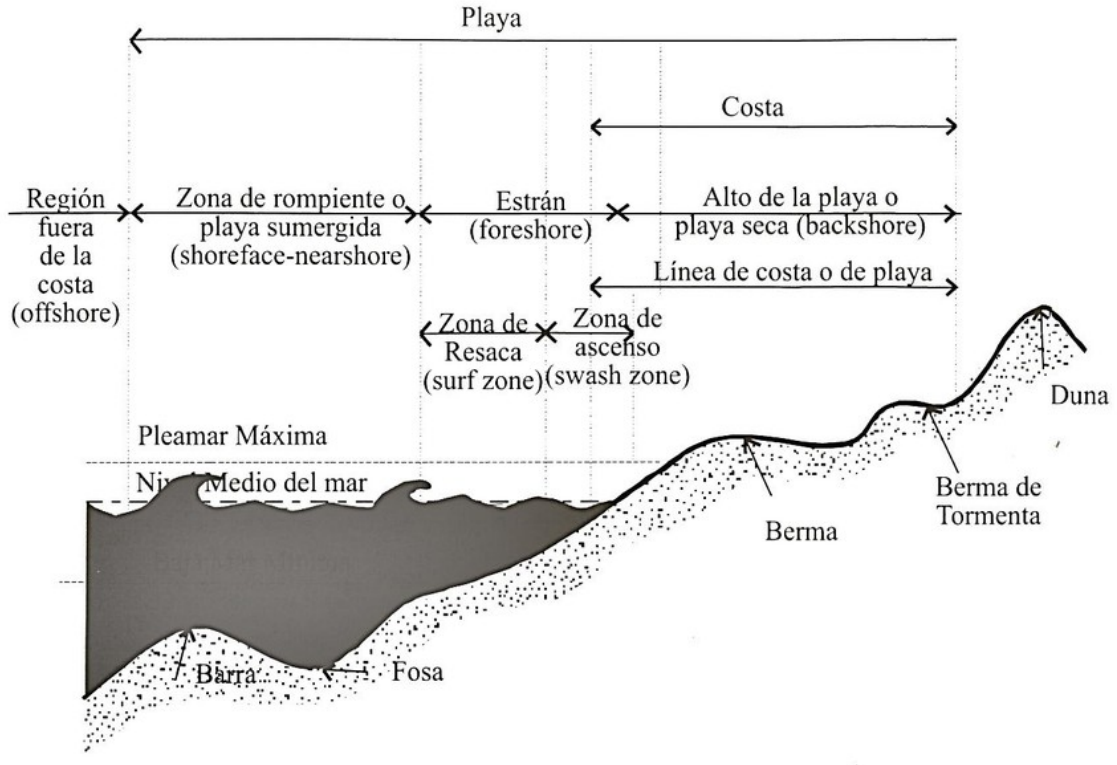
Objetivos

Que el estudiante profundice en:

1. El diseño de programas “escalables” basados en concurrencia por hilos y memoria compartida.
2. La programación, depuración y comprobación con MPI4PY en Python.
3. La evaluación del desempeño de un programa con MPI4PY en Python utilizando conceptos simples como “tiempo pared”, “aceleración” y eficiencia.

Descripción del problema

El conteo de tortugas en una arribada no es fácil por las condiciones en que usualmente debe realizarse (de noche) y por lo masivo de las mismas. Por esta razón se utilizan tres métodos de muestreo conocidos en adelante como: método del transecto paralelo a la berma (TPB), método de los cuadrantes (C) y el método de los transectos verticales sobre la berma (TVB). Para aclarar el término transecto y berma, considérese la siguiente figura:



Obsérvese que los niveles de las tres fases de la marea se ubican por debajo de la “berma”: este montículo es por tanto producto de la acción de la marea. Obsérvese que en una tormenta el nivel de la marea puede subir y generar una “berma de tormenta” más alto. Finalmente, no se debe confundir estas bermas con la duna eólica que se forman por la acción del viento y no de la marea.

Los transectos son franjas sobre la playa, ya sea el horizontal o paralelo a la berma, o los verticales ubicados perpendicularmente a la berma, entre ésta y las dunas eólicas. Cada cierta cantidad de minutos se repite el conteo de las tortugas en las áreas correspondientes, lo que se entiende como un muestreo. De acuerdo con esto, los métodos de muestreo se describen a continuación:

1. El TPB consiste en contar las tortugas en el transecto que inicia en la línea de costa (ver la figura, inicia aproximadamente en el punto donde llega la marea media) y se extiende en un ancho típico de 15 metros por una longitud de hasta un kilómetro. Un contador para este tipo de muestreo recorre toda la playa, se detiene 5 minutos y vuelve a recorrerla hasta que termine la simulación.

2. El TVB consiste en contar las tortugas que están en franjas verticales o perpendiculares a la berma y sobre ésta. Estos transectos típicamente miden dos metros (aproximadamente lo que mide una persona de mano a mano con los brazos extendidos al máximo hacia los lados) de ancho y su longitud va desde la berma hasta las dunas eólicas. El muestreo se hace cada cierta cantidad de minutos, se pueden incluir desde dos hasta cinco transectos cada cien metros de playa y no deben escrutarse más de 20 en un mismo muestreo.

3. El C consiste en contar las tortugas que están en cuadrantes previamente identificados y ubicados por encima de la berma. El muestreo se aplica cada cierta cantidad de minutos.

El problema de simulación consiste en modelar una arribada de N tortugas y los tres métodos de muestreo con el propósito de determinar cuál puede ser el mejor método y bajo qué condiciones es el más adecuado. Se trata entonces de realizar K experimentos de simulación cada uno con N tortugas arribando, y comparar los promedios de cada una de las estimaciones correspondientes a cada tipo de muestreo con el valor N, para intentar encontrar el mejor método de muestreo.

El objetivo fundamental en nuestro proyecto (además de la implementación correcta de la simulación) es reducir al máximo el “tiempo pared” (tp), así como experimentar con un proceso por cada sector de playa según el archivo “terreno.csv” y con diferente cantidad de procesos por nodo en algún cluster. Para tal efecto se deberán realizar los siguientes experimentos (en “experimentos.csv”):

Experimento de desempeño #1:
Cantidad de simulaciones: 10.
Línea del archivo marea.csv donde aparece el total de minutos a simular: 0.
Cantidad de tortugas: 70 mil.
Cantidad de procesos: uno por cada sector de playa.
Cantidad de procesos por nodo: 1,2,3.

Experimento de desempeño #2:
Cantidad de simulaciones: 10.
Línea del archivo marea.csv donde aparece el total de minutos a simular: 0.
Cantidad de tortugas: 700 mil.
Cantidad de procesos: uno por cada sector de playa.
Cantidad de procesos por nodo: 1,2,3.

Experimento de desempeño #3:
Cantidad de simulaciones: 10.
Línea del archivo marea.csv donde aparece el total de minutos a simular: 0.
Cantidad de tortugas: 7 millones.
Cantidad de procesos: uno por cada sector de playa.
Cantidad de procesos por nodo: 1,2,3.

Cuadro comparativo de desempeño que se deberá generar:

# ppn	Experimento #1			Experimento #2			Experimento #3		
1	tp_{11}	$ac_{11} = tp_{11}/tp_{11}$	$e_{11} = ac_{11}$	tp_{12}	$ac_{12} = tp_{12}/tp_{12}$	$e_{12} = ac_{12}$	tp_{13}	$ac_{13} = tp_{13}/tp_{13}$	$e_{13} = ac_{13}$
2	tp_{21}	$ac_{21} = tp_{11}/tp_{21}$	$e_{21} = ac_{21}/2$	tp_{22}	$ac_{22} = tp_{12}/tp_{22}$	$e_{22} = ac_{22}/2$	tp_{23}	$ac_{23} = tp_{13}/tp_{23}$	$e_{23} = ac_{23}/2$
3	tp_{31}	$ac_{31} = tp_{21}/tp_{31}$	$e_{31} = ac_{31}/3$	tp_{32}	$ac_{32} = tp_{22}/tp_{32}$	$e_{32} = ac_{32}/3$	tp_{33}	$ac_{33} = tp_{23}/tp_{33}$	$e_{33} = ac_{33}/3$

Debe interpretarse los tp_{ij} como los promedios de los “tiempo-pared” (ppn == i y experimento j) de las 10 diez simulaciones en cada experimento. El tiempo pared sólo debe contabilizar el tiempo neto de la simulación, omitiendo el tiempo para cargar datos y generar la salida.

Datos de entrada de la simulación

1. El archivo “terreno.csv” con información sobre los sectores en que se divide la playa sobre la cual se realizará la simulación. Por cada sector este archivo contiene una fila. En cada fila aparecen 4 números separados por comas:
1.1 longitud en metros del sector,
1.2 distancia del nivel de marea media a la berma en metros,
1.3 la altura de la berma respecto del nivel de marea media,
1.4 distancia de la berma a las dunas en metros.

2. El archivo “marea.csv” con información tres ciclos de aproximadamente 6 horas cada uno. Contiene tres filas cada una con 3 números separados por comas:
2.1 altura de la marea baja o alta en metros,
2.2 altura de la marea alta o baja en metros,
2.3 cantidad de minutos del ciclo a simular (nunca es exactamente 6 horas).

3. El archivo “comportamiento_tortugas.csv” contiene información sobre el comportamiento estocástico de las tortugas. En la primera fila:
3.1 0.2: probabilidad de desactivarse después de “vagar”,
3.2 0.2: probabilidad de desactivarse después de “camar”,
3.3 0.6: probabilidad de desactivarse después de “excavar”,
3.4 0.2: probabilidad de desactivarse después de “poner”,
3.5 0.01: probabilidad de desactivarse después de “tapar”,
3.6 0.01: probabilidad de desactivarse después de “camuflar”,
3.7 7.3: velocidad promedio estimada,
3.8 1.0: desviación estándar de la velocidad,
3.9 1.0: parámetro s (escala) para la distribución logística de la arribada. El parámetro μ debe ser igual a cero.

En la segunda fila la duración promedio y desviación estándar en minutos de cada estado posterior a “vagar”, de izquierda a derecha:
3.10 “camar”: promedio 1.58, desviación estándar 1.44
3.11 “excavar”: promedio 12.35, desviación estándar 4.92
3.12 “poner”: promedio 11.64, desviación estándar 4.34
3.12 “tapar”: promedio 4.98, desviación estándar 3.74
3.13 “camuflar”: promedio 5.01, desviación estándar 1.81
Se debe suponer que para cada estado de cada tortuga se debe generar un número aleatorio siguiendo la distribución normal con los parámetros correspondientes y que, aún así, puede que la tortuga decida a último momento desactivarse según las probabilidades de la primera línea.

En la tercera fila las probabilidades de anidamiento en las diferentes secciones de la playa, de izquierda a derecha:
3.14 0.17 probabilidad de anidar entre 0 y 10 metros antes de la berma,
3.15 0.53 probabilidad de anidar entre 0 y 10 metros después de la berma,
3.16 0.24 probabilidad de anidar entre 11 y 20 metros después de la berma,
3.17 0.06 probabilidad de anidar más allá de los 20 metros encima de la berma.

4. El archivo “cuadrantes.csv” con información sobre la aplicación del método C y los cuadrantes a usar. Contiene una primera fila con dos datos: la cantidad de contadores asignados y cada cuántos minutos se repetirá el conteo. Luego contiene una fila por cada cuadrante. En cada fila aparecen 4 números separados por comas:
4.1 coordenada X de la esquina inferior izquierda,
4.2 coordenada Y de la esquina inferior izquierda,
4.3 coordenada X de la esquina superior derecha,
4.4 coordenada Y de la esquina superior derecha.

5. El archivo “transectos_verticales.csv” con información sobre la aplicación del método TVB y los transectos verticales ubicados entre la berma y las dunas. Contiene una primera fila con dos datos: la cantidad de contadores asignados y cada cuántos minutos se repetirá el conteo. Luego contiene una fila por cada transecto vertical. En cada fila aparecen tres números:
5.1 coordenada X de la esquina inferior izquierda (X_{inf}),
5.2 coordenada Y de la esquina inferior izquierda,
5.3 coordenada Y de la esquina superior derecha.
5.4 La coordenada X de la esquina superior derecha NO se incluye pues es $X_{inf} + 2$.

6. El archivo “transecto_paralelo_berma.csv” con información sobre la aplicación del método TPB y el transecto horizonal paralelo a la berma. Contiene una primera fila con:
6.1.1 la cantidad de contadores asignados,
6.1.2 cada cuántos minutos se repetirá el conteo y
6.1.3 rango de visión hacia adelante en metros del contador.
La segunda fila:
6.2.1 el ancho del transecto en metros,
6.2.2 la longitud del transecto en metros.

Resultados o salida de la simulación

1. Cantidad total de tortugas N.
2. Cantidad de tortugas que efectivamente anidaron y por ende ovopositaron A.
3. Estimación por TPB de N.
4. Estimación por TVB de N.
5. Estimación por C de N.

NOTA: para hacerlas comparables TODAS las estimaciones se refieren a las tortugas que arribaron, sin importar en qué estado fueron vistas.

Tipos de agentes de la simulación

Esta es una simulación basada en dos tipos de agentes: “tortuga” y “contador”. Toda simulación basada en agentes requiere además un objeto “Simulador” que ejecute la simulación. De acuerdo con lo anterior usted deberá programar las siguientes clases:

Nombre de la clase	Función que cumple
Tortuga	Representar los atributos y el comportamiento de las tortugas.
Contador	Representar los atributos y el comportamiento de los contadores.
Simulador	Ejecutar la simulación

Se le provee un código base para cada clase que usted podrá modificar agregando atributos privados, métodos públicos y privados. Si es necesario modificar o eliminar algún método provisto se hará en acuerdo con el docente.

Las funciones del programa main() son:

1. Cargar y validar los datos del archivo “experimentos.csv”.
2. Cargar y validar los archivos de datos de entrada.
3. Ejecutar cada experimento indicado en el archivo “experimentos.csv”:
3.1 Asignar la instancia de Simulador con los datos de entrada.
3.2 Ejecutar la simulación invocando Simulador::simular(...).
3.3 Generar el archivo con los resultados de la simulación.
3.4 Generar la tabla de desempeño del experimento (3x3).

Escalas del modelo de simulación

1. Cada tic de la simulación representará un minuto.
2. Cada posición (X,Y) representará un metro cuadrado.
3. Cada tortuga ocupa una posición de un metro cuadrado.

Reglas que rigen el comportamiento de los agentes y la marea

El comportamiento de las tortugas pasa por la secuencia de estados indicada en "EstadoTortuga": {vagar, camar, excavar, poner, tapar, camuflar, inactiva} en ese orden empezando por "vagar". La tortuga puede desactivarse sin completar toda la secuencia, para lo cual se usan las probabilidades correspondientes en "comportamiento_tortugas.csv".

La velocidad de cada tortuga se determinará al azar usando la distribución normal (ver en la documentación de scipy: https://docs.scipy.org/doc/numpy/reference/generated/numpy.random.normal.html) con los parámetros de promedio y desviación estándar dados en el archivo correspondiente.

La posición inicial de cada tortuga en la playa se determinará al azar usando una distribución uniforme (ver https://docs.scipy.org/doc/numpy/reference/generated/numpy.random.uniform.html) para ambas coordenadas X,Y, tomando en cuenta que: 1) la coordenada Y puede variar entre la línea de la marea baja y la altura de la marea en el tic en que está entrando la tortuga a la playa y 2) las probabilidades descritas en 3.14 a 3.17.

La posición de anidamiento de la tortuga también se determinará al azar con base en las probabilidades de la tercera línea del archivo "comportamiento_tortugas.csv". Si una tortuga llega a anidar donde otra ya lo está haciendo, la segunda en llegar entra en estado "inactiva". Ninguna tortuga anidará más de 10 metros antes de la berma. Ninguna anidará más allá de las dunas.

El comportamiento de los contadores pasa por el ciclo indicado en "EstadoContador": {contar, esperar}, lo que implica que estos dos estados se repiten varias veces durante una simulación para cada contador. Si los contadores se desplazan lo hacen todos a velocidad constante de 6 kilómetros por hora.

El comportamiento de la arribada se modelará en un periodo de tantos minutos como se indique en la primera línea del archivo "marea.csv", contados a partir de la marea baja hasta la alta y mediante una distribución logística (ver https://en.wikipedia.org/wiki/Logistic_distribution) que alcanza su pico en la marea media. Los parámetros para la distribución vienen en el archivo "comportamiento_tortugas.csv"

El comportamiento de la altura de la marea se modelará por medio de una función lineal:

a = p*t + c

donde "a" es la altura en metros y "t" el tiempo en minutos. Para calcular "p" y "c", deben usarse los valores del archivo "marea.csv", bajo el supuesto de que por ejemplo, según la primera línea de dicho archivo, cuando t = 0, a debe ser 0.6 y cuando t = 372 a debe ser 2.2. Cada línea del archivo de mareas por tanto tendrá sus propios valores p y c, PERO SÓLO SE USARÁ LA PRIMERA LÍNEA en todos los experimentos.

Fórmula de cálculo para cada tipo de muestreo

Cada tipo de muestreo implica una fórmula de cálculo del total estimado de tortugas.

Mecanismo de conteo y fórmula de estimación para el método TPB

$T_e = N_c*i/(4.2*m)$, en la que:

1. T_e es el total estimado de tortugas que arribaron,
2. N_c es la sumatoria de la cantidad contada de tortugas en cada muestreo,
3. i es la cantidad de minutos entre muestreos, lo que dura caminando e inactivo,
4. m es la cantidad de muestreos que se aplicaron durante el periodo simulado.

Para el TPB es necesario simular el desplazamiento de los contadores, cuya cantidad aparece en el archivo correspondiente, a una velocidad de 6 kms por hora o 100 mts por minuto. La longitud total del transecto paralelo es de 1.5 kms.

Mecanismo de conteo y fórmula de estimación para el método TVB

$T_e = (A*d/(2*w*m*\sum l_j))*(N_c/pt)$

1. T_e es el total estimado de tortugas que arribaron,
2. A el área de observación total en metros cuadrados (entra la berma y las dunas),
2. d es la duración en minutos del fenómeno simulado,
3. w es el ancho en metros de cada transecto,
4. m es la cantidad de muestreos que se aplicaron durante el periodo simulado,
5. $\sum l_j$ es la sumatoria de las longitudes de todos los transectos,
6. N_c es la sumatoria de la cantidad contada de tortugas en cada muestreo,
7. pt promedio en minutos de cuánto duran las tortugas en el transecto: 64.8 mins.

En el archivo correspondiente aparecen 30 transectos verticales y un contador por cada transecto, dos cada 100 mts, para completar toda la playa a simular que, según el archivo “terreno.csv”, es de 1.5 kms.

Mecanismo de conteo y fórmula de estimación para el método C

$T_e = N_c*1,25*(A_c/A)*d/(1.08*m)$

1. T_e es el total estimado de tortugas que arribaron,
2. N_c es la sumatoria de la cantidad contada de tortugas en cada muestreo,
3. A_c es el área de un cuadrante, se supone que todos son del mismo tamaño,
4. A el área de observación total en metros cuadrados (entra la berma y las dunas),
7. d es la duración en minutos del fenómeno simulado,
8. m es la cantidad de muestreos que se aplicaron durante el periodo simulado.

NOTA: 1,25 porque aproximadamente un 25% de T_e se estima que anidan debajo de la berma. En el archivo correspondiente (“cuadrantes.csv”) aparecen los datos de seis cuadrantes para una playa de 1500 metros.

Criterios de evaluación o rúbrica de evaluación

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

1. **Escalabilidad demostrada**: su trabajo deberá funcionar correctamente y distribuir adecuadamente la carga con un proceso por sector de playa y tantos procesos por nodo como se ha indicado en el cuadro comparativo de desempeño.
2. **Desempeño demostrado**: su trabajo será comparado con aquél que muestre el **mejor desempeño por simulación** tomando en cuenta los experimentos descritos.
3. **Eficacia demostrada**: su trabajo deberá generar todas las salidas indicadas.
4. Eficiencia en el uso de memoria, basado en las estructuras de datos utilizadas.
5. Simplicidad del código.
6. Forma y estilo del código: código estilo “snake_case” el más utilizado con Python, nombres de clases empiezan en mayúsculas, nombres de métodos empiezan en minúscula, comentarios para los atributos y variables de métodos.
7. División de responsabilidades entre main-modelo: el main() sólo se ocupa de la entrada de datos, generar mensajes de error, el despliegue de ciertos resultados por la consola, invocación a los demás objetos para que realicen todos los procesamientos necesarios.

Fecha de entrega: domingo 7 de julio a las 23:55 por medio del enlace en el sitio del curso. SÓLO DEBERÁ SUBIR LOS ARCHIVOS DE CÓDIGO FUENTE (*.py) los archivos de salida de los tres experimentos y la tabla comparativa de desempeño.

Criterio	%Prc
Escalabilidad (requiere que el programa funcione)	25
Desempeño (requiere que el programa funcione)	35
Eficacia demostrada	40
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.
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.