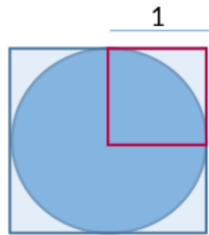


Trabajo Grupal Final

Cálculo de π con el Método de Simulación de Montecarlo

Conociendo las fórmulas para el cálculo del área del cuadrado y el círculo, y usando un generador de datos aleatorios uniforme, es posible calcular el valor de pi (π).

Imagínense un cuadrado de lado 2 unidades de medida y un círculo inscrito, que tendría entonces un radio de 1 unidad de medida:



El área de ese cuadrado es entonces de $2 \times 2 = 4$ unidades cuadradas, y el del círculo de $\pi * 1^2 = \pi$.

Consideremos un sistema orto normal, cuyo origen coincide con el centro del círculo. Tenemos entonces la ecuación del círculo: $x^2 + y^2 \leq 1$. Todo punto de coordenadas (x_p, y_p) del cuadrante superior derecho que se encuentre dentro del círculo tendrá como propiedad $x_p^2 + y_p^2 \leq 1$.

Si solo tenemos en cuenta ese cuadrante, y obtenemos aleatoriamente puntos dentro de ese cuadrante siguiendo una distribución uniforme, todos los puntos estarán dentro del área de ese cuadrante del cuadrado, pero no todos estarán dentro del área del círculo. Pero al ser el cuadrado y el círculo simétricos, y al tener el mismo baricentro, las proporciones del área del círculo y del área del cuadrado se conservan en el cuadrante.

De manera general, se observa la siguiente proporción:

$$\frac{\pi * l^2}{\text{puntos en el círculo}} \rightarrow \frac{(2 * l)^2}{\text{puntos totales}}, \text{ y despejando } \pi, \text{ obtenemos:}$$

$$\pi = 4 * \frac{l^2 * \text{puntos en el círculo}}{l^2 * \text{puntos totales}}$$

El objetivo de este trabajo es diseñar un sistema de software que calcule π a partir de tirajes aleatorios uniformes tal y como lo acabamos de describir. Este método se conoce como simulación de Montecarlo, y es una forma estática de simulación.

Indicaciones

1. Diseñe e implemente un sistema de software cliente-servidor para calcular π de manera secuencial con componentes SCA (usando FraSCAti).

2. En una segunda versión, modifique el sistema para que el usuario de su sistema pueda especificar, usando una GUI: (número de puntos aleatorios a generar, *seed* del generador aleatorio a utilizar, número de nodos de procesamiento iniciales a utilizar. El sistema deberá realizar el conteo secuencial de los puntos y el cálculo de π , mostrando en la misma GUI el valor resultante.
3. El sistema también deberá permitir recibir un archivo plano CSV con las configuraciones correspondientes a varios intentos de cálculo de π (nombre de configuración, número de puntos aleatorios a considerar, *seed* del generador aleatorio a utilizar, número de nodos de procesamiento a utilizar). Las soluciones deberán escribirse en un archivo de salida, con una línea por resultado (nombre de configuración, resultado de π).

Deberá utilizar una implementación de un generador pseudo aleatorio uniforme, que le permita inicializar una semilla (*seed*), de tal forma que pueda reproducir los resultados encontrados anteriormente si así lo requiriese. Pueden utilizar la clase `java.util.Random` con su método `nextDouble` para tal efecto.

Primera versión

La primera versión debe correr con un solo nodo de cálculo (versión simplificada).

Deberá entregar en un archivo zip como sigue:

CalculoPi-V1-ApellidosIntegrantes.zip

- Archivo Word con la explicación de la solución, incluyendo el diagrama de deployment
- Archivo `readme.txt` con el nombre de los integrantes, instrucciones de compilación, instrucciones de ejecución de los componentes.
- Código fuente de la solución realizada y composite FraSCaTi

Segunda versión

Rediseñe la solución de la versión anterior para que el cálculo de π sea distribuible, además de escalable en tantos componentes FraSCaTi como sea necesario. Para tal propósito, debe seguir considerando el cuadrante superior derecho, pero debe poder particionarlo en varias regiones. El conteo de los puntos de cada región será la responsabilidad de un nodo FraSCaTi diferente. Puede que una misma máquina física se encargue primero de una región, envíe resultados a un nodo central, y luego se encargue de una o más regiones si todavía hacen falta regiones por cubrir.

Para propósitos prácticos, su solución debe distribuirse en al menos 4 nodos de procesamiento.

Deberá entregar en un archivo zip como sigue:

CalculoPi-V2-ApellidosIntegrantes.zip

- Documentos
 - o Archivo Word con la explicación de la solución, incluyendo un diagrama de deployment, especificando la estructura de procesamiento (SISD, SIMD, MISD, MIMD) y la estructura de almacenamiento que usa su arquitectura (UMA, NUMA, NORMA).

- o Archivo readme.txt con el nombre de los integrantes, instrucciones de compilación, instrucciones de ejecución de los componentes.
- TipoNodoProcesamiento1
 - o Src
 - resources (archivos .composite FraSCAti, archivo readme con instrucciones para ejecutar)
 - java (código fuente de clases e interfaces Java de los composite del folder respectivo)
 - o clases (código compilado de clases e interfaces Java, incluyendo .jars resultantes)
- ...
- TipoNodoProcesamientoN
 - o Src
 - resources (archivos .composite FraSCAti, archivo readme con instrucciones para ejecutar)
 - java (código fuente de clases e interfaces Java de los composite del folder respectivo)
 - o clases (código compilado de clases e interfaces Java, incluyendo .jars resultantes)

Para ambas versiones:

- Archivo de test con 5 entradas para 5 cálculos independientes, considerando 1,000,000, 100,000,000, 10,000,000,000, 1,000,000,000,000, 100,000,000,000,000 puntos generados, con los *seeds* 1, 2, 3, 10 y 15 respectivamente.
- Archivo de salida correspondiente con los resultados experimentales del cálculo de las 5 entradas, cada uno ejecutados 10 veces y tabulados con los respectivos tiempos de respuesta, y sus promedios, y por el número de nodos de procesamiento.

No se recibirán trabajos por otro medio diferente a Moodle, sin excepción.