

UNIVERSIDAD DEL VALLE DE GUATEMALA
Colegio Universitario



Propuesta de aplicación que utiliza paralelismo mediante hilos con
OpenMP.

Sofia Lopez
Adrián López
Javier Chávez

Programación de Microprocesadores

Temática:

Simulaciones.

LINK DE GITHUB: (Contiene source file .c del programa y su versión secuencial para la comparación)

<https://github.com/adirnnn/Proyecto2-Microprocesadores-OpenMP-Threads.git>

FASE 1**Descripción del problema:**

La temática consiste en predecir la rentabilidad de una inversión de una empresa mediante simulaciones Monte Carlo haciendo uso de dos procedimientos, uno con integrales y el otro con cadenas de Markov. El programa realizaría ambos procedimientos en paralelo basándose en datos históricos proporcionados para proyectar un resultado probable el cual será comparado para proporcionar un rango de incertidumbre. Tanto el resultado probable como la incertidumbre establecerían el riesgo del proyecto.

Parte secuencial:

La lectura y el preprocesamiento de los datos de entrada se realizan de manera secuencial para garantizar que todos los datos estén disponibles antes de iniciar las simulaciones.

La configuración de parámetros y variables necesarios para las simulaciones.

Parte paralela:

La generación de números aleatorios para las simulaciones se realizaría en paralelo mediante hilos privados.

Ambas simulaciones (mediante integrales y cadenas de Markov) se ejecutarán en paralelo, dividiendo las iteraciones de los bucles entre múltiples hilos.

Los resultados parciales de cada hilo se combinan utilizando la directiva de reducción para obtener los resultados finales.

Justificación del uso de paralelismo:

Al dividir las simulaciones entre múltiples hilos, la aplicación puede aprovechar de mejor manera los recursos del hardware, reduciendo significativamente el tiempo de ejecución de las simulaciones.

La paralelización permite que la aplicación escale eficientemente con el número de núcleos de la CPU, manejando un mayor número de simulaciones en menos tiempo.

Al paralelizar las tareas más intensivas computacionalmente, se minimiza el tiempo de espera para obtener los resultados, permitiendo decisiones más rápidas y basadas en mayores datos si es deseado.

Análisis:

- ¿Qué acciones debe poder realizar el programa?

Debe de ser capaz de analizar una gran cantidad de datos y escenarios pasados para luego utilizarlos como base para sus cálculos.

Debe de ser capaz de realizar dos procedimientos complejos en paralelo (integrales y cadenas de Markov), para determinar un resultado final por procedimiento.

Debe de ser capaz de comparar ambos resultados para determinar un rango de incertidumbre y de esa manera proporcionar una estimación más precisa.

- ¿Qué variables y tipos de datos serán utilizados?
 - Nombre de cada inversión: Identificador (String value) para diferenciarlas, consiste en el nombre del asset invertido.
 - Rentabilidad: Consiste en la tasa de retorno periódica de la inversión (float value) (puede ser diaria, mensual o anual; en nuestro caso será mensual).
 - Plazo: Duración del período de la inversión. (Int value) (puede ser valor diario, mensual o anual; en nuestro caso será mensual).
 - Volatilidad: Indicador de clave de riesgo, es un porcentaje de probabilidad de casos extremos que será representado en decimal. (float value).
 - Flujo de caja: Egreso e ingreso de efectivo por ingreso. (Int).
 - Valor en riesgo: Cuantificación del riesgo asociado con el ingreso a un mercado mediante inversiones. (Float value).

$$VaR = - P (\sigma \sqrt{T}) Z$$

P : valor de la inversión o precio del activo

σ : desviación estándar, que puede ser estimada con cualquiera de los métodos anteriores

T : horizonte de tiempo analizado

Z : número de desviaciones estándar de la distribución normal estándar correspondiente al nivel de confianza deseado.

- ¿Qué información necesitará pedir al usuario?

Las variables previamente listadas sobre cada inversión pasada y las posibles del negocio al que se desea invertir, sobre el cual se realizarán las simulaciones.

- ¿Qué cálculos o procesos deben ser paralelizados?

Cálculo de Monte Carlo mediante integrales.

Cálculo de Monte Carlo mediante cadenas de Markov.

Diseño:

Pseudocódigo:

//Parte secuencial

Obtener los datos históricos

Inicializar parámetros

- Def número de simulaciones N
- Def un estado inicial (x_o)
- Def plazo P
- Def límites de integración a, b
- Def f(x)

Generación de número aleatorios $i=1 \dots =N$

Promedio $f(xi) \rightarrow S = \sum_{i=1}^N f(xi)$

Varianza $V = \frac{1}{N} \sum_{i=1}^N (f(xi) - promedio)^2$

```
#pragma omp parallel sections
```

```
//Parte paralela:
```

```
#pragma omp section {
```

```
    - Cadena de Markov
```

```
Crear matriz de transición T
```

```
function cadenadeMarkov(N, P, a, b);
```

```
    for (int i = 0; i < N; i++) {
```

```
        resultado = 1.0;
```

```
        for (int t = 0; t < P; t++) {
```

```
            Cambiar de estado basado en la matriz
```

```
            Generar retorno basado en el estado actual
```

```
            Actualizar el valor de la cartera
```

```
        Promedio de los resultados (RF)
```

```
        return RF
```

```
    }
```

```
#pragma omp section {
```

```
    - Integral
```

```
Estimación valor integral  $I \approx \frac{(b-a)S}{N}$ 
```

```
//Pseudo
```

```
function MonteCarloIntegral(N, a, b, f):
```

```
    S = 0
```

```
    for i = 1 to N do:
```

```
        x_i = random(a, b)
```

```
        S = S + f(x_i)
```

```
    end for
```

```

    I = (b - a) * S / N
    return I
}
Calcular incertidumbre
Calcular el nivel de riesgo
Mostrar resultados

```

Catálogo de las funciones desarrolladas para implementar el algoritmo de solución:

- Estado getNextState(Estado estadoActual):
 - Propósito: Simula la transición entre los diferentes estados (alto, bajo, estable) basándose en la matriz de probabilidades de transición y retorna el siguiente estado de la inversión basado en la matriz de transición.
- double generateNormal(double mean, double std):
 - Propósito: Genera un número aleatorio siguiendo una distribución normal (gaussiana) usando el método de Box-Muller y retorna un valor aleatorio basado en la distribución normal especificada.
- double monteCarloIntegracion(int numPuntos, double invInicial):
 - Propósito: Realiza una simulación Monte Carlo utilizando la integración numérica para calcular la rentabilidad de la inversión y retorna la rentabilidad calculada como porcentaje.

Resultados obtenidos:

```
PS D:\28adr\OneDrive\Documents\Uni\Ciclo 4\Programación de Microprocesadores\proyecto2fase2micro> ./simulacionesMonteCarlo
Resultado de la simulación con integrales: 74.46%
Rentabilidad Promedio con Cadenas de Markov: 49.74%
Rentabilidad Final Combinada: 62.10%
Incertidumbre entre ambos métodos: 24.72%
```

Tal como se observa, los resultados obtenidos de parte del programa consisten en el resultado de la simulación Monte Carlo mediante integrales, el resultado de la simulación Monte Carlo mediante cadenas de Markov, la rentabilidad final combinada de ambos y la incertidumbre entre resultados.

Viéndolo desde la perspectiva empresarial, la cual es la perspectiva de la temática seleccionada como problemática, el resultado de la simulación con integrales es un resultado de perspectiva optimista sin consideración mayor de los riesgos de mercado o la volatilidad; el resultado con la simulación con cadenas de Markov es un resultado pesimista con las variables de riesgo adicionales, la rentabilidad final combinada ya se puede observar como un resultado realista de la inversión y finalmente, la incertidumbre sirve para observar la importancia de la consideración de las variables de riesgo a la hora de considerar la inversión.

Comparación de Eficiencia entre Paralelo y Secuencial:

Para realizar una comparación entre nuestro programa en paralelo y el programa en secuencial, creamos una versión puramente secuencial del mismo programa y le agregamos a ambos el return del tiempo de ejecución. Para la versión secuencial es un clock y para la paralela es un omp get time.


```

PS D:\28adr\OneDrive\Documents\Uni\Ciclo 4\Programación de Microprocesadores\proyecto2fase2micro> ./simulaciones
Resultado de la simulación con integrales: 74.46%
Rentabilidad Promedio con Cadenas de Markov: 49.74%
Rentabilidad Final Combinada: 62.18%
Incertidumbre entre ambos modelos: 24.72%
Tiempo total de ejecución (sec): 0.01
PS D:\28adr\OneDrive\Documents\Uni\Ciclo 4\Programación de Microprocesadores\proyecto2fase2micro> ./simulacionesSecuencial
Resultado de la simulación con integrales: 74.40%
Rentabilidad Promedio con Cadenas de Markov: 49.74%
Rentabilidad Final Combinada: 62.07%
Incertidumbre entre ambos modelos: 24.67%
Tiempo total de ejecución (sec): 0.02
PS D:\28adr\OneDrive\Documents\Uni\Ciclo 4\Programación de Microprocesadores\proyecto2fase2micro>

```

Tal como se puede observar, bien es cierto que el tiempo total no es mayor, puede ser la velocidad de procesamiento de 8 núcleos o el hecho de que por demostración no estamos utilizando demasiados datos históricos como para crear un delay significativo; de todas maneras, se puede observar una clara diferencia donde la versión secuencial toma el doble de tiempo que la versión paralela en proporcionar los resultados. Por lo que se cumple el argumento inicial de que la paralelización es una mejora del programa al correr ambos procesos simultáneamente en lugar de uno luego del otro.

Planificación:

Actividad de Desarrollo	Meta	Fecha Programada
Establecimiento de Temática/Problema	Definir el área que se abordará para resolver una problemática relacionada.	Martes 6 Agosto
Descripción Detallada del Problema	Identificar un problema dentro de la temática elegida y buscar la forma de	Martes 6 Agosto

	poderlo abordar.	
Definición de Sección Paralela/Secuencial	Identificar la parte del código que será realizada de forma secuencial y las tareas que se ejecutarán de forma paralela.	Miércoles 7 Agosto
Justificación de Paralelismo	Comprender y explicar la importancia y utilidad de utilizar el paralelismo en nuestra solución.	Miércoles 7 Agosto
Análisis de Temática	Lograr definir las diferentes funciones que tendrá el código, identificar las variables que se utilizarán y los cálculos que se llevarán a cabo, al igual que la información que se necesitará del usuario.	Miércoles 7 Agosto
Pseudocódigo/Diagrama de Flujo	Identificar la estructura y el orden que tendrá el código y representar cómo se llevarán a cabo las distintas funciones	Martes 13 Agosto
Cronograma de Planificación	Organizar los días en los que se realizará el proyecto	Martes 6 Agosto

Referencias

Claro, A. (2014, January 14). *Metodos de monte carlo en mecánica estadística*. SlideShare.

<https://www.slideshare.net/slideshow/metodos-de-montecarlo-en-mecanica-estadistica/30002681#2>