

## Taller de Evaluación de Rendimiento

### 1. Objetivos principales

- Comparar algoritmos y sus versiones entre serie (secuencial) y paralelo.
- Comparar diferentes sistemas de cómputo con SO Linux.
- Analizar e interpretar los resultados para extraer recomendaciones y conclusiones.

### 2. Preparación

- Seleccionar 2 sistemas de cómputo con SO Linux: Nativo, máquina virtual PUJ, COCALC, etc.
- Descomprimir la carpeta enviada (archivos fuentes).
- Documentar los archivos (explicar brevemente qué hace cada función)
- Verificar el funcionamiento correcto de los programas (que compilan y se ejecutan).
- Documentar el archivo en lenguaje PERL ([lanzador.pl](#)).
- Se recomienda (opcional) hacer un Makefile, como archivo para automatizar la compilación.

### 3. Evaluación de Rendimiento

- Compilar los archivos fuente.
- Ejecutar para valores de matrices menores a 5x5, usando el comando:

```
$ ./ejecutableMM tamMatriz numHilos
```

Donde:

ejecutable: nombre del ejecutable

tamMatriz: Dimensión de matriz a multiplicar

numHilos: cantidad de hilos

Ejemplo por consola de ejecución para tamaño 4x4 y numero de hilos 1

```
$ ./ejecutableMM 4 1
```

Una vez comprobado que todo funciona satisfactoriamente, se puede observar que la salida representa los tiempos de ejecución del algoritmo clásico de multiplicación de matrices para una dimensión de matriz (tamMatriz) y un número de hilos (NumHilos) determinado. La siguiente parte corresponde a tomar medidas de rendimiento. Al evaluar el rendimiento se tiene que tener en cuenta que diversos factores del sistema operativo, hacen que los valores de tiempo varíen para los mismos argumentos (dimensión y número de hilos). Para poder capturar una idea sobre el valor real del tiempo, es necesario aplicar la teoría de la probabilidad, bajo el término genérico de ley de los grandes números, la cual engloba varios teoremas que describen el comportamiento del promedio de una sucesión de variables aleatorias conforme aumenta

su número de ensayos. Lo anterior representa que se debe hacer una batería de experimentación para capturar el valor promedio, y así determinar un valor cercano a la realidad.

### Batería de experimentación

Se recomienda hacer un diseño de experimentos, el cual tenga en cuenta la cantidad de hilos a comparar, y la carga máxima o dimensión de matriz. Lo anterior, en función de las configuraciones de los sistemas de cómputo a comparar. A continuación, se da una lista de consideraciones pertinentes.

- Seleccionar los sistemas de cómputo a comparar y presentar las características de hardware correspondientes.
- Seleccionar valores diferentes de tamaño de matrices (`tamMatriz`), teniendo en cuenta la concurrencia a aplicar y la jerarquía de memoria (justificar).
- Seleccionar valores de hilos de ejecución (`NumHilos`) justificar.
- Ejecutar por cada valor seleccionado al menos 30 veces.
- Elaborar una tabla que represente la batería de experimentación a realizar, para hacer el análisis de rendimiento del algoritmo de Multiplicación de Matrices, en serie (1 hilo) y en paralelo (2,4,8,... hilos).
- Ejecutar la batería de experimentos de forma automatizada. Para automatizar el proceso de ejecución se debe hacer uso del archivo en PERL. El archivo (`lanzador.pl`) que anteriormente han documentado, presenta los diferentes argumentos que usted puede cambiar para hacer que se automatice la captura de todas las ejecuciones de su batería de experimentación. Para ejecutar los experimentos:

`./lanzador.pl`

Siéntase libre de modificar el archivo `lanzador.pl`, para que pueda capturar los valores deseados y para cada argumento obtener el promedio. El archivo `lanzador.pl` entrega para cada ejecución (cada 30 repeticiones) un archivo tipo (.dat) el cual puede exportar a cualquier aplicación “hoja de cálculo” de su preferencia (office, excel, numbers, etc) y allí poder hacer los análisis correspondientes, incluido gráficas que enfaticen o mejoren su análisis comparativo.

## 4. Rúbrica de Evaluación

Se evaluará el informe en formato PDF (cualquier otro formato tendrá calificación 0), el cual ha de subir al BS. Adicional al documento en PDF, debe subir una carpeta con los archivos documentados en el proceso (.c, .h y .pl). La carpeta debe estar comprimida en formato zip. En el documento deberá presentar (introducción y sección inicial):

- Métricas que permitan evaluar el desempeño de los programas comúnmente utilizados, y cuál métrica de desempeño usó en el taller.
- Descripción de las plataformas de Hw y Sw utilizadas para realizar las medidas.

Las tablas 1 y 2 muestran el detalle de la rúbrica de evaluación.

Indicadores de Desempeño	Excelente	Bueno	Deficiente
Diseño de experimentos basado en el análisis de los sistemas de cómputo y la concurrencia de los procesos e hilos. Manejo apropiado de los recursos.	El desarrollo del taller es según el enunciado, respeta requerimientos de entrada, salida y todas las especificaciones para la ejecución. Elabora y sigue el diseño de experimentos.	Se resuelve el taller siguiendo las especificaciones dadas, pero tiene fallas menores, básicamente en el diseño de experimentos (dimensiones de matrices, concurrencia, etc)	No se siguen las indicaciones dadas para resolver el taller, no hay diseño de experimentos o no se sigue el diseño de experimentos.
Análisis de algoritmos y documentación en las fuentes, de forma clara. Se presenta en forma modular. Documentación satisfactoria.	El código presenta documentación y estructura satisfactoria. Se ha tomado en cuenta las verificaciones de los algoritmos. Cierra archivos, elimina recursos	El código presenta documentación con algunas deficiencias. Presenta estructura y hace verificaciones de los algoritmos. Pequeñas fallas en la documentación	Se presenta alguna falla importante en 2 o más de los siguientes aspectos: <ul style="list-style-type: none"> <li>● Documentación</li> <li>● Verificación de los algoritmos.</li> <li>● Manejo adecuado de</li> </ul>
Indicadores de Desempeño	Excelente	Bueno	Deficiente
	temporales, entre otros.	o poca documentación. Manejo adecuado de los recursos.	los recursos.
Presentación de informe que cubra el desarrollo del taller, el cual incluya el diseño de experimentos.	Se presenta un informe completo estructura satisfactoria. Adicional, se presenta análisis de resultados usando la estadística, imágenes, gráficas; para las comparativas. Realiza conclusiones y recomendaciones, según los resultados obtenidos.	Presenta informe incompleto, con errores de gramática y ortografía. Los análisis de resultados son pobres y/o no presenta tablas ni gráficas que sustentan las conclusiones y recomendaciones.	El estudiante no realiza el ejercicio.

**Tabla 1.** Indicadores de desempeño del proyecto Simulador de Reservas de Parques.

Actividad	Porcentaje	Excelente	Bueno	Deficiente
Diseño de experimentos basado en el análisis de los sistemas de cómputo y la concurrencia de los procesos e hilos. Manejo apropiado de los recursos.	3%	1.5	1.0	<0.75
Análisis de algoritmos y documentación en las fuentes, de forma clara. Se presenta en forma modular. Documentación satisfactoria.	3%	1.5	1.0	<0.75
Presentación de informe que cubra el desarrollo del taller, el cual incluya el diseño de experimentos.	4%	2.0	1.0	[0.0,0.8]
<b>Total</b>	<b>10%</b>	<b>5</b>	<b>3</b>	

**Tabla 2.** Guía de valoración para cada uno de los niveles de cada rúbrica.