

Trabajo Práctico Nº 1

Optimización de algoritmos secuenciales

Fecha límite para el envío: viernes 9 de abril

Pautas generales:

- La entrega es en grupos de a los sumo dos personas.
- Los algoritmos deben ser ejecutados sobre el cluster provisto por la cátedra:
 - En IDEAS se encuentra el instructivo que explica cómo usarlo.
 - Mediante mensajería, debe solicitar las credenciales de acceso (si aun no lo hizo).
- Se recomienda desarrollar en sus máquinas locales y usar el tiempo del cluster para las pruebas de producción.
- El tiempo de ejecución debe obtenerse sólo de la parte del algoritmo que realiza el procesamiento. Por lo tanto, NO debe incluir:
 - Alocación y liberación de memoria
 - Impresión en pantalla (printf)
 - Inicialización de estructuras de datos
 - Impresión y verificación de resultados
- El envío de los archivos debe realizarse por mensajería de IDEAS a los docentes Enzo Rucci y Adrian Pousa. Se debe enviar:
 - Los archivos .c con el código fuente de cada ejercicio.
 - Un informe en PDF que describa brevemente las soluciones planteadas, análisis de resultados y conclusiones. El informe debe incluir el detalle del trabajo experimental (características del hardware y del software usados, pruebas realizadas, etc), además de las tablas (y posibles gráficos, en caso que corresponda) con los tiempos de ejecución.
- La entrega cuenta con un coloquio por el medio virtual establecido por la cátedra.

Enunciado:

1. Resuelva el ejercicio 6 de la Práctica Nº 1 usando dos equipos diferentes: (1) cluster remoto y (2) equipo *hogareño* al cual tenga acceso (puede ser una PC de escritorio o una notebook).
2. Desarrolle un algoritmo en el lenguaje C que compute la siguiente ecuación:

$$C = T + avg_R(RA + RB)$$

Donde A , B , C , T y R son matrices cuadradas de $N \times N$.

avg_R es el valor promedio de los elementos de la matriz R .

El elemento (i,j) de la matriz R debe calcularse como:

$$R_{i,j} = (1 - T_{i,j})(1 - \cos\theta_{i,j}) + T_{i,j}\sin\theta_{i,j}$$

Donde $T_{i,j}$ es el elemento en la posición (i,j) de la matriz T .

El ángulo $\theta_{i,j}$, en radianes, se obtiene de la posición (i,j) de una matriz M de $N \times N$. Los valores de los elementos de la matriz M están comprendidos en un rango entre 0 y 2π .

Mida el tiempo de ejecución del algoritmo en el cluster remoto. Las pruebas deben considerar la variación del tamaño de problema ($N=\{512, 1024, 2048, 4096\}$). Por último, recuerde aplicar las técnicas de programación y optimización vistas en clase.

Información útil:

Sobre las funciones matemáticas: Para trabajar con las funciones trigonométricas se deberá incluir la biblioteca matemática de GCC:

```
#include <math.h>
```

Sobre la generación de datos: para generar números aleatorios en punto flotante seguir los siguientes pasos:

Incluir las bibliotecas necesarias:

```
#include<time.h>
#include<stdlib.h>
```

En el main inicializar el generador aleatorio:

```
int main(int argc, char* argv[]){
    time_t t;
    srand((unsigned) time(&t));
    ...
}
```

Utilizar la siguiente función que genera un número aleatorio en un rango dado:

```
double randFP(double min, double max) {
    double range = (max - min);
    double div = RAND_MAX / range;
    return min + (rand() / div);
}
```

Sobre la constante π : para generar aleatoriamente los valores de la matriz M se requiere la constante π que debe ser definida en el código de la siguiente forma:

```
#define PI 3.14159265358979323846
```

Luego, cada elemento de M puede obtenerse invocando a la función:

```
randFP(0, 2*PI)
```