

MANUAL DE USUARIO

MP Programming and false sharing

By: Gerald Mora Mora, Jonathan

Ejecución automática del proyecto:

Paso 1: Entrar a la carpeta donde está contenido el proyecto.

Paso 2: Ejecutar el terminal desde esa localización.

Paso 3: Verificar que exista el archivo makerun.

Paso 4: Ejecutar el siguiente comando:

```
$> chmod +x makerun
```

Paso 5: Ejecutar el siguiente comando para lanzar el makerun:

```
$> ./makerun
```

Paso 6: Primero se mostrarán los resultados gráficos de la solución con C++, por lo que debes darle al botón de guardar imagen.

Paso 7: Cerrar ventana de Python.

Paso 8: Posteriormente se mostrarán los resultados de la solución en C, debes de igual manera dar clic en el botón de guardar imagen.

Paso 9: Cerrar ventana de Python.

Paso 10: En el terminal estará impresa información relevante del benchmark también.

Compilacion y ejecucion individual:

Para Caso en C++:

SingleThread:

```
$> g++ -o mSST mSST.cpp -pthread -std=c++11 -lbenchmark
```

```
$> ./mSST
```

Caso con False Sharing:

```
$> g++ -o mSFS mSFS.cpp -pthread -std=c++11 -lbenchmark
```

```
$> ./mSFS
```

Caso 1 sin False Sharing:

```
$> g++ -o mSNFS1 mSNFS1.cpp -pthread -std=c++11 -lbenchmark
```

```
$> ./mSNFS1
```

Caso 2 sin False Sharing:

```
$> g++ -o mSNFS2 mSNFS2.cpp -pthread -std=c++11 -lbenchmark
```

```
$> ./mSNFS2
```

Para caso en C:

SingleThread:

```
$> gcc -o mSSTC mSSTC.c -lpthread  
$> ./mSSTC
```

Caso con False Sharing:

```
$> gcc -o mSFSC mSFSC.c -lpthread  
$> ./mSFSC
```

Caso 1 sin False Sharing:

```
$> gcc -o mSNFS1C mSNFS1C.c -lpthread  
$> ./mSNFS1C
```

Caso 2 sin False Sharing:

```
$> gcc -o mSNFS2C mSNFS2C.c -lpthread  
$> ./mSNFS2C
```

Perfilado de programas con Cachegrind para C++ y C respectivamente:

SingleThread:

```
$> valgrind --tool=cachegrind ./mSSTC 2> mSSTCCPP_results.txt  
$> valgrind --tool=cachegrind ./mSSTC 2> mSSTCC_results.txt
```

Caso con False Sharing:

```
$> valgrind --tool=cachegrind ./mSFSC 2> mSFSCCPP_results.txt  
$> valgrind --tool=cachegrind ./mSFSC 2> mSFSCC_results.txt
```

Caso 1 sin False Sharing:

```
$> valgrind --tool=cachegrind ./mSNFS1 2> mSNFS1CCPP_results.txt  
$> valgrind --tool=cachegrind ./mSNFS1C 2> mSNFS1CC_results.txt
```

Caso 2 sin False Sharing:

```
$> valgrind --tool=cachegrind ./mSNFS2 2> mSNFS2CCPP_results.txt  
$> valgrind --tool=cachegrind ./mSNFS2C 2> mSNFS2CC_results.txt
```

Perfilado de programas con Perf:

SingleThread:

```
$> perf stat -e cpu-cycles -e L1-dcache-loads -e L1-dcache-load-misses -e  
L1-dcache-stores -e L1-dcache-store-misses -e LLC-load-misses -e  
LLC-store-misses -e cache-references -e cache-misses -e branch-misses -e  
mem-loads -e mem-stores -e instructions ./mSSTC
```

Caso con False Sharing:

```
$> perf stat -e cpu-cycles -e L1-dcache-loads -e L1-dcache-load-misses -e
L1-dcache-stores -e L1-dcache-store-misses -e LLC-load-misses -e
LLC-store-misses -e cache-references -e cache-misses -e branch-misses -e
mem-loads -e mem-stores -e instructions ./mSFSC
```

Caso 1 sin False Sharing:

```
$> perf stat -e cpu-cycles -e L1-dcache-loads -e L1-dcache-load-misses -e
L1-dcache-stores -e L1-dcache-store-misses -e LLC-load-misses -e
LLC-store-misses -e cache-references -e cache-misses -e branch-misses -e
mem-loads -e mem-stores -e instructions ./mSNFS1C
```

Caso 2 sin False Sharing:

```
$> perf stat -e cpu-cycles -e L1-dcache-loads -e L1-dcache-load-misses -e
L1-dcache-stores -e L1-dcache-store-misses -e LLC-load-misses -e
LLC-store-misses -e cache-references -e cache-misses -e branch-misses -e
mem-loads -e mem-stores -e instructions ./mSNFS2C
```

Instalación de herramientas:

Perf:

```
$> sudo apt install linux-tools-common
```

```
$> sudo apt-get install linux-tools-5.8.0-50
```

Valgrind:

```
$> sudo apt-get install valgrind
```