

# Manual de Usuario de False Sharing Inspection

## Introducción

Bienvenido al Manual de Usuario de False Sharing Inspection. Este manual proporciona una guía rápida para comenzar a utilizar nuestro software. Si tienes alguna pregunta o necesitas ayuda adicional, no dudes en contactarnos en [correo electrónico de soporte].

## Requisitos del Sistema

Asegúrate de que tu sistema cumpla con los siguientes requisitos antes de usar False Sharing Inspection:

Sistema Operativo: Linux (distribuciones basadas en Ubuntu)

Procesador: AMD o Intel

Memoria RAM: Sin requisitos de memoria

Espacio en Disco Duro: 300 Kb (En caso de querer hacer las pruebas con Gem5+Ruby se requiere de 11 Gb)

## Paquetes necesarios para la ejecución del programa en Linux:

- **git (Git):** `sudo apt install git`
- **gcc 7+:** `sudo apt install build-essential`
- **perf:** `apt-get install linux-tools-common linux-tools-generic linux-tools-`uname -r``
- **Python3:** `sudo apt install python3-dev`
- **Valgrind:** `sudo apt install valgrind`

Si desea hacer uso de de gem5 y Ruby ocupa instalar también:

- **SCons 3.0+:** `sudo apt install scons`
- **protobuf 2.1+ (Optional):** `sudo apt install libprotobuf-dev protobuf-compiler libgoogle-perftools-dev`
- **Boost (Optional):** `sudo apt install libboost-all-dev`

## Instalación

Siga estos pasos para instalar False Sharing Inspection:

Descarga el repositorio: [https://github.com/papmora/Proyecto1\\_Arqui2](https://github.com/papmora/Proyecto1_Arqui2)

Una vez descargado el repositorio puede lanzar el ejecutable de Python para hacer uso de la aplicación. Por lo que debería de ver la siguiente interfaz:



En caso de querer hacer uso de gem5 siga estos pasos:

Clone o descargue el repositorio: <https://github.com/gem5/gem5>

Puede hacer uso del comando: `git clone https://github.com/gem5/gem5`

Luego teniendo en cuenta que debe hacer uso del protocolo MSI para esta prueba ejecute los siguientes comandos dentro del directorio gem5:

- `scons build/X86_MSI/gem5.opt --default=X86 PROTOCOL=MSI SLICC_HTML=True`
- `build/X86_MSI/gem5.opt configs/learning_gem5/part3/simple_ruby.py`  
**En este caso dentro del archivo simple\_ruby.py se puede observar el benchmark que se va a ejecutar, por lo que se debe garantizar que esa dirección es la del benchmark.**
- `build/X86_MSI/gem5.opt --debug-flags=ProtocolTrace`  
`configs/learning_gem5/part3/ruby_test.py | grep 0x4ac0`

## Funcionalidades Principales

False Sharing Inspection ofrece las siguientes funcionalidades principales:

Funcionalidad 1: Ejecución de test bench single y multi threading en C

Funcionalidad 2: Ejecución de test bench single y multi threading en C++

Funcionalidad 3: Análisis de gráficas con resultados del test bench.

Funcionalidad 4: Análisis de Hardware objetivo.

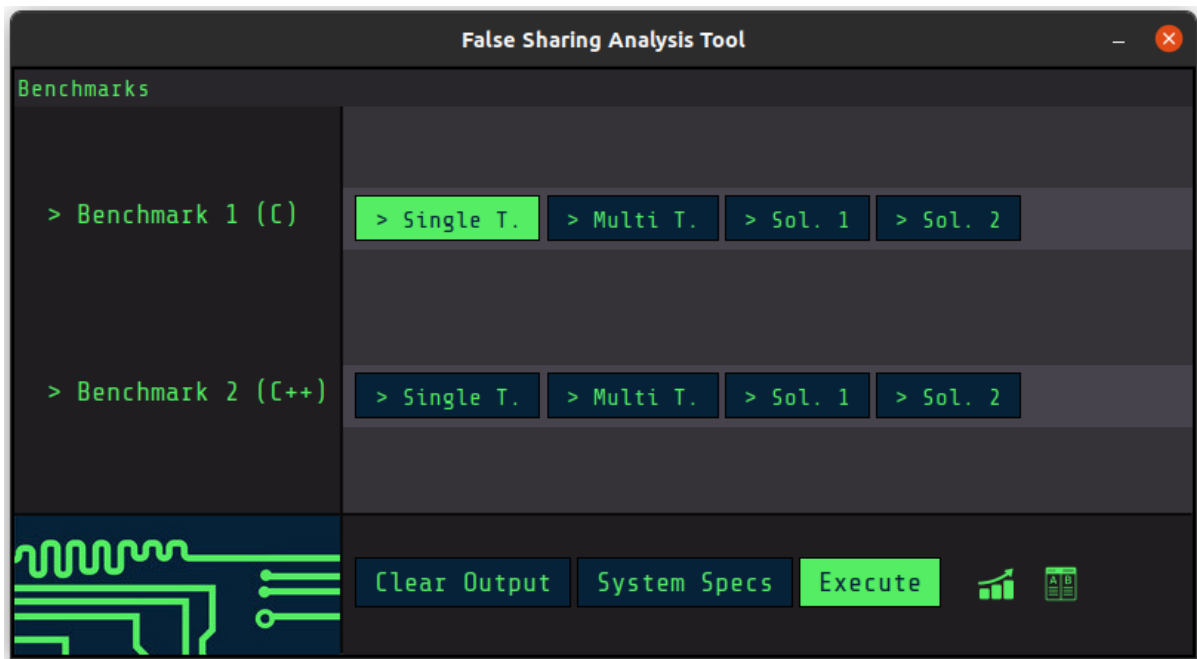
Funcionalidad 5: Comparativa de gráficas.

Funcionalidad 6: Limpiar los archivos txt que contienen las salidas.

## Cómo Utilizar False Sharing Inspection

Funcionalidad 1

Seleccione la opción que desea en la parte superior como se muestra en la imagen y luego haga clic en el botón ejecutar.

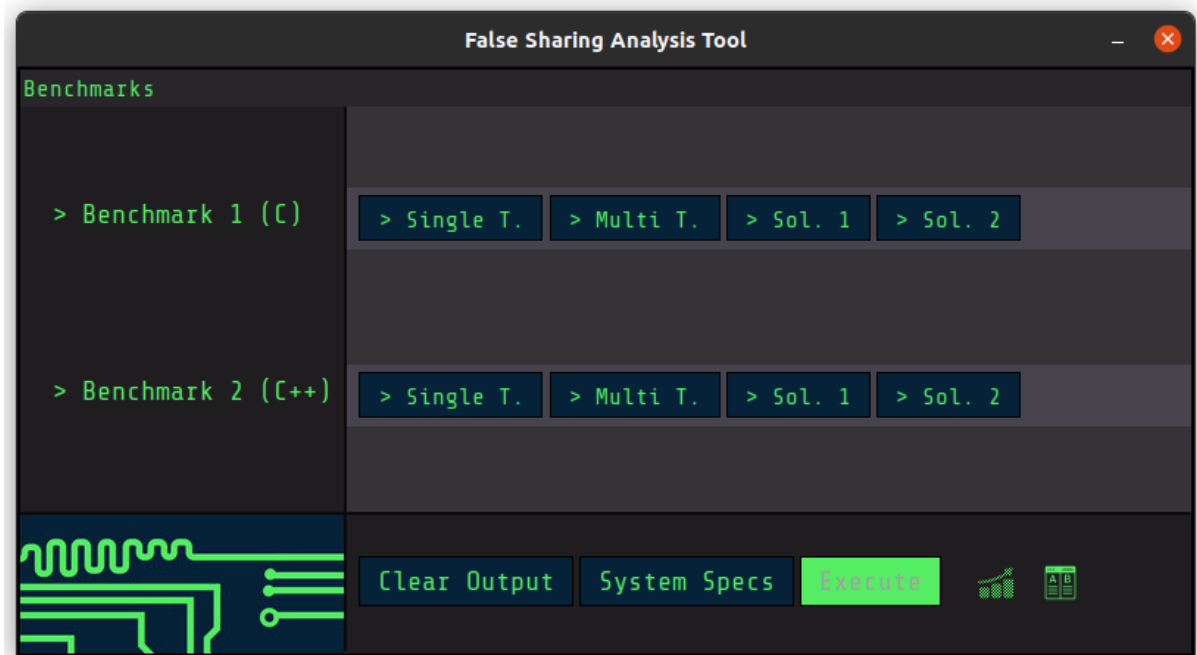


#### Funcionalidad 2

Al igual que la anterior, pero esta vez en el donde indica Benchmark 2, seleccione la opción que desea y luego haga clic en el botón ejecutar.

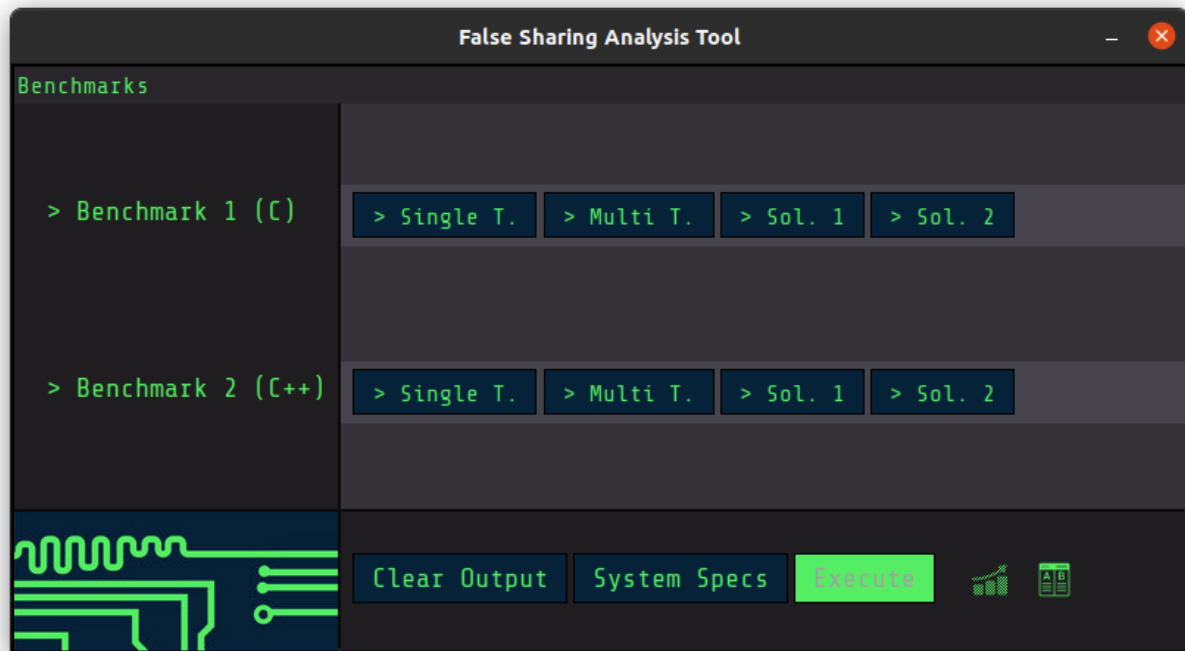
#### Funcionalidad 3

Haga click en el botón de la parte inferior que tiene icono de gráfica



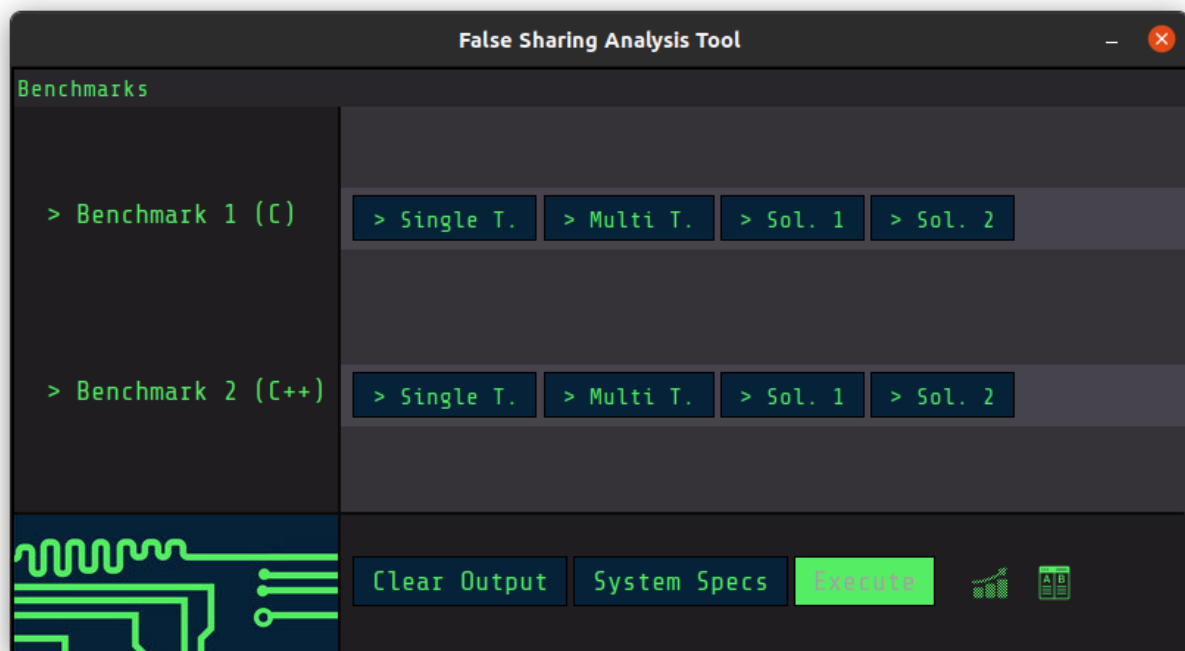
#### Funcionalidad 4

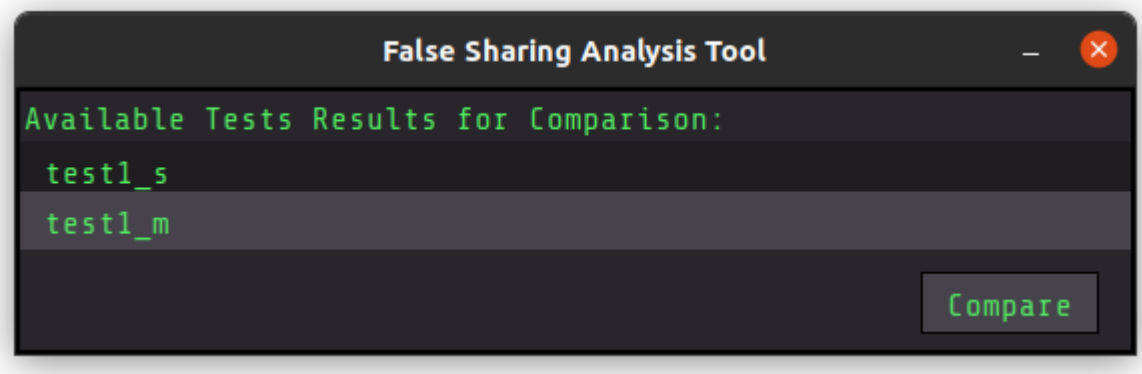
Haga click en el botón de la parte inferior que tiene icono que dice System Specs



#### Funcionalidad 5:

Haga click en el último icono de la parte de abajo a la derecha, este le mostrará los benchmark que cuentan con datos para analizar y comparar las gráficas. Seleccione los que desea comparar y haga clic en Compare.





Funcionalidad 6

Haga click en el botón de la parte inferior que tiene icono que dice Clear Outputs

## Preguntas Frecuentes

P: ¿Los gráficos no muestran nada y los archivos txt solo tienen signo de dólar?

R: En estos casos asegúrese que cuenta con perf instalado en su sistema operativo.

Asimismo verifique que tiene los permisos del kernel para poder ejecutar perf haciendo uso de estos comandos en la terminal:

Acceso temporal:

```
sudo sysctl -w kernel.perf_event_paranoid=-1
```

Verificación:

```
cat /proc/sys/kernel/kptr_restrict
```

Si Verification = 1 en la terminal:

```
sudo sh -c 'echo 0 > /proc/sys/kernel/kptr_restrict'
```

Si desea restablecer el acceso temporal:

```
sudo sysctl -w kernel.perf_event_paranoid=2
```

Si desea restaurar el valor del kernel:

```
sudo sh -c 'echo 1 > /proc/sys/kernel/kptr_restrict'
```

Si desea acceso permanente:

- Añadir al archivo /etc/sysctl.conf : kernel.perf\_event\_paranoid = -1
- Comando: sudo nano /etc/sysctl.conf

## Contacto

Si tienes alguna pregunta o necesitas asistencia, no dudes en contactarnos al correo:  
mora.pablojavier@gmail.com

Gracias por utilizar False Sharing Inspection. ¡Esperamos que tengas una experiencia excelente!