

Plan de Pruebas de Software

Información del Proyecto

Nombre del Proyecto: False Sharing Inspection

Equipo de Desarrollo:

- Pablo Mora
- Roy Acevedo
- Daniel Camacho

Objetivo del Proyecto

El objetivo de este proyecto es hacer un análisis de los sistemas MP cuando surge el efecto de false sharing e implementar medidas que mitiguen este efecto en el sistema.

Objetivos del Plan de Pruebas

El objetivo de este plan de pruebas es garantizar la calidad y el funcionamiento correcto del software False Sharing Inspection. Se describirán las actividades de prueba que se llevarán a cabo, los recursos necesarios y los criterios de aceptación.

Alcance de las Pruebas

Las pruebas se centrarán en las siguientes áreas del software:

- Correcto funcionamiento del backend (benchmarks, recolección de datos de ejecución)
- El frontend debe tener una interfaz de usuario intuitiva.

Ambiente de Pruebas

Se utilizará una computadora con linux (precisamente Ubuntu) y arquitectura x86 para realizar las pruebas.

Tipos de Pruebas:

Funcionales

- Debe ser capaz de ejecutar cada uno de los benchmark un mínimo de 10 veces. Esto con el objetivo de contar con suficiente información para hacer el análisis.

Criterios de Aceptación: Se considerará la prueba exitosa cuando el archivo de output de cada uno de los benchmark cuente con la información de mínimo 10 ejecuciones.

- Debe ser capaz de recolectar la información de cada iteración de la ejecución de los benchmark, para poder disponer de esa información para hacer la comparativa entre benchmark.

Criterios de Aceptación: Se considerará la prueba exitosa si cada benchmark cuenta con un archivo de output independiente que recolecta cada una de sus iteraciones.

- Debe ser capaz de crear la gráfica que muestre el resultado de las iteraciones de cada benchmark, para tener la posibilidad de visualizar con mayor rapidez el impacto o resultado de cada benchmark.

Criterios de Aceptación: Se considerará la prueba exitosa si se logra observar el resultado de las iteraciones de forma gráfica permitiendo mostrar el impacto del benchmark.

- Debe ser capaz de comparar gráficas de los diferentes benchmark, para poder ejemplificar el single vs multi threading de cada benchmark, así como el resultado de las soluciones de cada benchmark.

Criterios de Aceptación: Se considerará la prueba exitosa si se logra observar la clara diferencia de forma gráfica, con respecto al impacto entre la ejecución del benchmark single o multi threading, así como los benchmark con solución que mitigue el efecto de false sharing.

- Debe ser capaz de analizar el hardware objetivo del equipo que ejecuta el programa, para poder personalizar los programas acorde a las especificaciones del sistema.

Criterios de Aceptación: Se considera exitosa esta prueba si se logra mostrar en pantalla las características del hardware objetivo.

No Funcionales

- El sistema debe tener una interfaz de usuario intuitiva que permita a los usuarios completar tareas sin confusiones.

Criterios de Aceptación: Se considerará la prueba exitosa si tras las pruebas realizadas el usuario puede hacer uso de la aplicación sin necesidad de consultar o tener algún tipo de confusión.

- El tiempo de ejecución de cada benchmark no debe superar los 3 minutos en una computadora con especificaciones gama media.

Criterios de Aceptación: Se considerará la prueba exitosa si al ejecutar esta prueba en diferentes computadoras de gama media, el tiempo de ejecución de cada benchmark no supera los 3 minutos.

Anexos

- Hennessy, J., & Patterson, D. (2017). Computer Architecture: A Quantitative Approach (6th ed.). Elsevier Science.
- 4- Sorin, J & Hil, Mark D. (2011) A Primer on Memory Consistency and Cache Coherence.