

Análisis del Rendimiento de Funciones C en Diferentes Arquitecturas y Compiladores

CI0131 - Diseño de Experimentos

Universidad de Costa Rica

May 7, 2025

Contenido

- 1 Descripción del Problema
- 2 Hipótesis y Objetivos
- 3 Variable de Respuesta
- 4 Unidad Experimental
- 5 Factores de Diseño
- 6 Factores Constantes
- 7 Factores Variables
- 8 Factores de Molestia
- 9 Restricciones y Diseño
- 10 Conclusión

Problema de Investigación

Contexto

En el dominio del desarrollo de software de alto rendimiento, la elección de la arquitectura y el compilador ejerce una influencia sustancial en el tiempo de ejecución y el uso de recursos.

Propósito

Cuantificar el impacto de estas variables y su relación a través de metodologías de diseño y análisis de experimentos, proporcionando información valiosa para la toma de decisiones en proyectos de ingeniería de software.

Hipótesis y Objetivos

Pregunta de Investigación

¿Cuál es el efecto principal de la arquitectura de la CPU y el compilador en el rendimiento de funciones en C?

Objetivos

- Medir el desempeño de ejecución de las funciones seleccionadas en diferentes combinaciones de arquitectura y compilador.
- Analizar la influencia de la arquitectura y el compilador en la variabilidad del desempeño de funciones.

Variable de Respuesta: Descriptivo

Nombre	Descripción	Justificación	Tipo	Unidades
T_exec	Tiempo transcurrido desde el inicio hasta el fin de la ejecución del binario compilado de la función	En esta variable almacenamos del tiempo de ejecución de la batería de pruebas	Continua	Milisegundos

Variable de Respuesta: Metodológico

Nombre	Metodología de recolección	Rango	Precisión y Exactitud
T_exec	Se ejecutará cada función con medición de tiempo utilizando herramientas como time, perf o el mismo clock del sistema operativo, Tras investigación exploratoria se decidirá si cada medición será repetida 5 veces para disminuir posibles sesgos, o solo 1 vez si vemos que no es factible repetirlo 5 veces.	0 ms – 100,000 ms	Exactitud: Repetir 100 ejecuciones de manera secuencial para cuantificar variabilidad. Precisión: calcular la desviación estándar de las ejecuciones anteriormente mencionadas.

Unidad Experimental

Nombre	Descripción	Justificación	Tipo	Unidades
Función C	Función escrita en lenguaje C que implementa una lógica específica y que será compilada y ejecutada.	La función en C es la unidad fundamental sobre la cual se medirá el rendimiento, ya que permite evaluar directamente el impacto de la arquitectura y el compilador sobre un código idéntico.	Discreta categórica	N/A

Factores de Diseño: Descriptivo

Nombre	Descripción	Justificación de inclusión	Justificación como factor de diseño	Tipo y unidades
Arquitectura CPU	Plataforma hardware utilizada para ejecutar el binario	Afecta el rendimiento por diferencias de architecture (ej: manejo de memoria)	Fundamental para probar la hipótesis de variabilidad entre plataformas	Discreta categórica (intel, amd)
Compilador	Herramienta utilizada para generar el binario de la función	Diferentes optimizaciones afectan el desempeño	Necesario para medir la influencia del compilador en el rendimiento	Discreta categórica (GCC, Clang)

Factores de Diseño: Metodológico - Arquitectura CPU

Nombre	Metodología	Rango	Niveles	Justificación	Precisión y exactitud
Arquitectura CPU	Uso de hardware real para cada arquitectura para eliminar el overhead de un emulador	intel i7, amd	2	Representan arquitecturas populares, relevantes y contemporáneas entre ellas	Precisión: cambiar físicamente de máquina limpia en ¡ 1 min. Exactitud: comprobar frecuencia nominal con lscpu y tolerancia $\pm 1\%$.

Factores de Diseño: Metodológico - Compilador

Nombre	Metodología	Rango	Niveles	Justificación	Precisión y exactitud
Compilador	Selección del compilador instalado apropiadamente para cada arquitectura	GCC, Clang	2	Ambos son cornerstones de los compiladores C opensource	Exactitud: validar con gcc --version y clang --version. Precisión: repetir compilación 5 veces (verificar durante EDA si es factible o no), verificar hash idéntico y tamaño ± 0 bytes.

Factores que se Mantendrán Constantes: Descriptivo

Nombre	Descripción	Justificación de inclusión	Justificación como factor constante	Tipo y unidades
Código fuente	Funciones escritas en C	Garantizar comparabilidad	Para atribuir diferencias únicamente a arquitectura o compilador	N/A

Factores que se Mantendrán Constantes: Metodológico

Nombre	Metodología de fijación	Rango	Nivel	Justificación	Precisión y exactitud
Sistema operativo	Se usará la misma versión del sistema operativo entre ejecuciones de pruebas	ubuntu 24.02	1	No es parte del experimento probar diferentes sistemas operativos	Chequeo previo con lsb_release cada día de prueba.
Versión del compilador e flags	Usar misma versión del compilador y mismo comando	gcc 15.1	1	Para aislar efecto del compilador puro sin variación de flags	Script automatizado que imprime versión y flags antes de cada corrida.
Cantidad de programas en primer plano	Solo estara ejecutandose las pruebas en primer plano	1	1	Lo ideal es tener la misma carga de trabajo en todas las pruebas	Medir uso de CPU y RAM antes de cada bloque; debe ser 5%

Factores a los que se Permitirá Variar

Nombre	Descripción	Justificación de inclusión	Justificación como factor variable	Tipo y unidades
Carga del sistema	Procesos en segundo plano y uso de recursos del sistema	La carga del sistema puede afectar los tiempos de ejecución	No se controlará para reflejar condiciones más realistas de uso en producción	Continua (% de uso de CPU/memoria)
Temperatura de CPU	Temperatura del procesador	La temperatura puede influir en el rendimiento	Representar condiciones reales de operación	Continua (°C)
Tamaño de datos	Volumen de datos procesados por la función	Puede afectar los patrones de acceso a memoria y caché	Se medirá su efecto pero no se controlará estrictamente	Discreta numérica (bytes)

Factores de Molestia

Nombre	Descripción	Justificación de inclusión	Estrategia de minimización	Tipo y unidades
Código con librerías dinámicas	Bibliotecas compartidas cargadas en tiempo de ejecución	Pueden variar la latencia de enlace dinámico	USar herramientas como prelink	Categórica (sí/no)
Interrupciones de CPU y cambios de contexto	Eventos de sistema que detienen el proceso y redistribuyen tiempo CPU	Alteran mediciones de tiempo de ejecución	Se registran con perf record y se aleatoriza el orden de pruebas	Continua (eventos/minuto)
Materiales y enfriamiento (throttling térmico)	Temperatura del procesador durante la ejecución	La temperatura puede influir en el rendimiento	Se permite variar naturalmente para representar condiciones reales	Continua (°C)

- Solo se disponen de dos máquinas físicas (Intel y AMD)
- No se probarán sistemas operativos ni flags adicionales
- Máximo de 5 ejecuciones por combinación debido a tiempo computacional (se tiene que averiguar durante EDA si es factible a nivel de tiempo)
- No se incluyen arquitecturas virtuales/emuladas

Aspectos de Diseño

- Diseño factorial 2^2 (2 arquitecturas \times 2 compiladores)
- Dos réplicas completas (10 ejecuciones por combinación)
- Aleatorización: orden aleatorio de las corridas

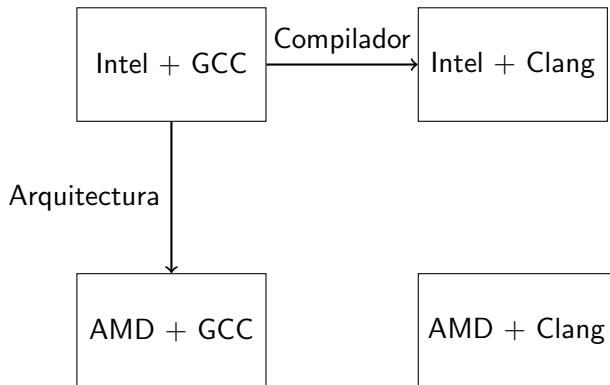


Figure: Diseño factorial 2^2

- Se realizara pruebas preliminares con 10 funciones de prueba para verificar scripts de automatización y medición (ademas de la EDA)
- Ajustes realizados:
 - Confirmación de sincronización de relojes
 - Limpieza de caché entre ejecuciones
- No hay datos de interés adicional para el experimento principal

Conclusión y Sigüientes Pasos

Valor esperado del estudio

Este experimento proporcionará información cuantitativa sobre:

- Efectos principales de arquitectura y compilador en el rendimiento
- Posibles interacciones entre estos factores
- Variabilidad del rendimiento bajo condiciones controladas

Aplicaciones prácticas

Los resultados pueden informar decisiones de:

- Selección de hardware para desarrollo de software
- Elección de compilador según el caso de uso
- Estimación de rendimiento en diferentes entornos

¡Gracias por su atención!
¿Preguntas?