

Instituto Tecnológico de Costa Rica Escuela de Ingeniería en Computadores Programa de Licenciatura en Ingeniería en Computadores Curso: CE-4302 Arquitectura de Computadores II Profesores: Luis Alonso Barboza Artavia Ronald García Fernández Semestre: I 2025	Taller 01 <i>Multithreading & Thread Level Speculation</i> Fecha de asignación : 18/03/25 (grupo 02) 19/03/25 (grupo 01) Fecha de entrega : 08/04/25 (grupo 02) 09/04/25 (grupo 01) Grupos de trabajo: 2 personas (máximo)
---	---

1. Descripción

En el desarrollo de aplicaciones que explotan el ***Thread Level Parallelism (TLP)*** en sistemas *multicore*, uno de los aspectos importantes a considerar es el manejo de *threads* respecto al manejo de recursos compartidos de HW y SW, este taller tiene como objetivo introducir al estudiante conceptos básicos para el análisis e implementación de software que emplea *multithreading* en [C++20](#) para modelar la técnica llamada *Thread Level Speculation*, además se pretende el uso básico de herramientas tipo ***Large Language Models (LLMs)*** para el análisis de código.

2. Requisitos

- a- Un computador con soporte *multithreading*
- b- Sistema operativo basado en Linux (no virtualizado)*
- c- GCC con soporte para C++20

3. Investigue

A continuación, se formulan una serie de preguntas de guía para investigar sobre la implementación de *multithreading* en C++20

- a- ¿Para la sirven las bibliotecas `std::atomic` `std::thread`?
- b- En C++20 ¿Cómo es posible establecer determinismo al utilizar funciones que generan valores pseudo aleatorios?
- c- ¿Cuál es la diferencia en `enum class` y `typedef enum` en C++?
- d- Si se utiliza `g++/gcc` como compilador ¿Cómo es posible especificar el uso de C++20?
- e- ¿Cómo es posible en C++20 determinar el número de hardware *threads* soportados en una plataforma objetivo?
- f- ¿Qué métodos se emplean para crear *threads* y para sincronizarlos?
- g- ¿En qué consiste el concepto de ***thread level speculation (TLS)*** en el contexto de arquitectura de computadores y sistemas *multicore/multithread*?
- h- ¿Cómo es posible utilizar ***Large Language Models (LLMs)*** para hacer análisis sintáctico y semántico de código?

4. Ejemplo de implementación de programas *multithreading*: modelado simplificado que soporta TLS

Para efectos de este taller se le brinda un [código](#) fuente que implementa un modelo simplificado de TLS se le pide analizar y responder siguiente:

- a- ¿Explique en qué consiste el modelo presentado (Elabore su respuesta basándose en la teoría del curso)?
- b- ¿Explique cuál es el propósito de las líneas 35, 43, 49, y 60 del archivo `tls.cpp` (Utilice herramientas tipo LLM y [la referencia del lenguaje](#))
- c- ¿Qué consideraciones y/o limitaciones encuentra respecto al número de **hardware threads** en el modelo mostrado (Justifique)?
- d- Basado en su respuesta del punto c, si su plataforma objetivo posee un número menor de **hardware threads** realice los cambios necesarios para que funcione.
- e- Compile el modelo proporcionado y ejecútelo varias veces muestre los resultados obtenidos, discuta los resultados.

5. Implemente

- a- Basándose en lo realizado en las secciones 3 y 4, implemente una versión del modelo de TLS llamada `tls_mod.cpp`, que cumpla con los siguientes requisitos:
 - i. Soportar determinismo a la hora de modelar retardos en la etapa **EXECUTE**.
 - ii. Soportar en tiempo de ejecución la selección del número de **HW threads**.
 - iii. Incluir más de una variable compartida (por ejemplo: `speculative_data_2`) de forma que los **HW threads** elijan aleatoriamente (con determinismo) una de ellas, **esto debe soportarse en tiempo de ejecución**.

6. Justifique

- a- Utilizando lo implementado en el punto 5, grafique el porcentaje de **successful commits** para diferentes valores de **HW threads** (sugerencia implemente un script para poder realizar esto) y diferentes cantidades de variables compartidas, debe incluir al menos 2 casos (1 variable compartida y más de una variable compartida, ver figura 1).

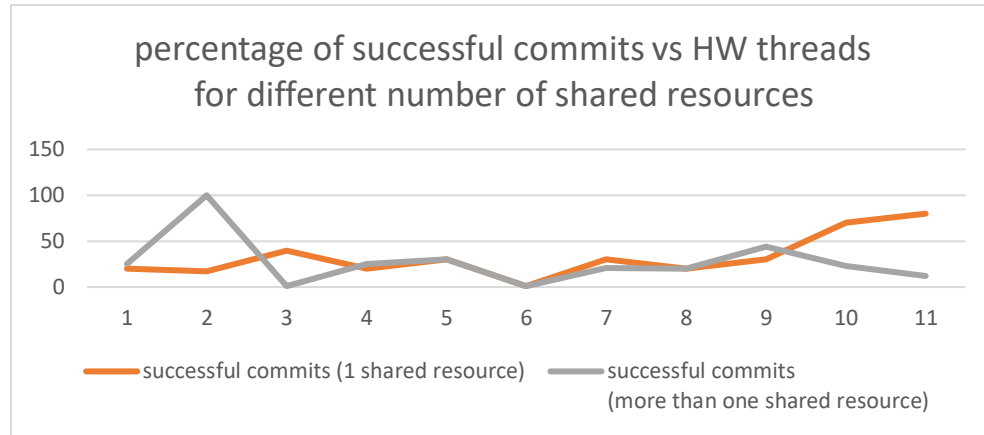


Figura 1. Ejemplo de gráfico para mostrar resultados

- b- Discuta sus resultados basándose en la teoría del curso y contraste que sucede al disminuir la probabilidad de que un recurso sea compartido en TLS y que sucede al aumentar el número de **HW threads**.
- c- ¿Qué puede concluir de la técnica TLS basado en su modelo modificado y los resultados obtenidos, particularmente sobre el resultado teórico máximo esperado?

7. Entregables

- a- Un documento **PDF** donde muestre las respuestas a las preguntas planteadas en secciones 3 y 4, junto con los resultados de la sección 5, 6, para las secciones en donde utilice *LLMs* debe presentar el análisis hecho sobre los resultados de la herramienta y la validación respectiva de los mismos.
- b- El código fuente con la implementación del punto 5 y el archivo README con los detalles para ejecutar su código y graficar sus resultados, NO incluya ejecutables de ser así se le penalizará con 15 puntos.

Si tienen dudas puede escribir al profesor al correo electrónico. Los documentos serán sometidos a control de plagios. La entrega se debe realizar por medio del TEC-Digital en la pestaña de evaluación.

No se aceptan entregas extemporáneas después de la fecha de entrega a las 23:59 como máximo.