***Ideas basicas de paralelismo***

***Sesión practica – Week 3***

*Samuel Francisco Moncayo Moncayo*

**1.Preguntas**

*A. Describir las cuatro clasificaciones de Flynn y explicar por que la configuración MISD es vista como no-sensata*

1. SISD (Single instruction, Single data)

* Un único procesador por lo tanto ejecuta una sola instrucción sobre un solo flujo de datos
* Es una arquitectura clásica y secuencial
* Utilizado en computadores antiguos

2. SIMD (Single instruction, Multiple data)

* Un unico flujo de instrucciones aplicado con diferentes flujos de datos en paralelo
* Útil en diferentes operaciones, como arreglos o operaciones vectoriales
* Utilizado en GPUs modernas

3. MISD (Multiple instruction, Single data) (No sensato)

* Varios procesadores ejecutan diferentes instrucciones con el mismo flujo de datos
* Este es considerado no sensato por que casi no existen aplicaciones prácticas, porque el mismo dato no suele necesitar múltiples operaciones diferentes al mismo tiempo.

4. MIMD (Multiple instruction, Multiple Data)

* Varios procesadores ejecutan instrucciones diferentes sobre datos diferentes al mismo tiempo
* Es la base de los multiprocesadores y multinúcleos

*B. ¿Qué limitaciones tiene el paralelismo a nivel de instrucción y por qué surgen?*

*r/* El paralelismo a nivel de instrucción se ve limitado por las dependencias en datos, control y recursos, ya que, no siempre muchas instrucciones se pueden ejecutar en paralelo si es que dependen unas de otras o compiten por los mismos recursos. Asimismo, la necesidad de un riguroso orden incrementa riesgos y complejidad, creando barreras físicas y de eficiencia

*C. ¿Qué tendencias en microarquitectura y macroarquitectura han llevado a que los programadores tengan que expresar explícitamente paralelismo? ¿Cuáles son los retos de exponer el paralelismo al software?*

*r/* Debido a que ya no es viable aumentar únicamente la velocidad de los procesadores o añadir mas micro transistores, las arquitecturas actuales se enfocan en añadir más núcleos y combinar CPU con aceleradores especiales. Esta nueva mecánica necesita que los programadores se familiaricen con el paralelismo dentro de sus aplicaciones. Sin embargo, estos nuevos enfoques representan retos, mayor complejidad al momento de programar, posibles condiciones de carrera, dificultades de escalabilidad o variaciones en el rendimiento según el hardware de la maquina.

*D. ¿Qué es el HyperThreading y cómo aprovecharlo desde el software?*

*r/* El HyperThreading es una tecnología diseñada por Intel que permite que solo un núcleo físico ejecute dos hilos, mejorando en sobremedida la utilización de los recursos del procesador. Generalmente ofrece ganancias del 20%-30%, no duplica el rendimiento por que ambos hilos comparten la mayoría de los recursos, sin embargo, los desarrolladores pueden escribir programas multihilo con un buen escalado, evitando asumir que cada núcleo lógico tiene la misma potencia que uno físico.