

# **CURSO: COMPUTACIÓN DE ALTO PERFORMANCE**

## **IGP Instituto Geofísico Peruano**

**Docente:** MS. Renzo Emilio Bustamante Avanzini

### Descripción de clases

#### **Sesión 1: Introducción a Sistemas Distribuidos, Sistemas de Computación y UNIX**

Se explicará el cambio del paradigma de trabajo de manera secuencial a paralelo y las principales motivaciones de este, se describirá como funciona un sistema de computación: CPU, Memoria Principal y Disco, haciendo una descripción de cómo funciona la memoria virtual y time sharing dentro de un Sistema Operativo. Middleware para Sistemas Distribuidos. Se discutirá a se analizar UNIX-LINUX, comandos básicos, permisos de carpetas, creación de cuentas de usuario, acceso de remoto, conexión segura. Ley de Moore y Ley de Amdalh.

#### **Sesión 2: Repaso de C**

En esta sesión de hará un repaso general de todo el lenguaje C, pasando por variables, condicionales, funciones, arreglos, cadenas, punteros, manejo del stack y el heap y archivos. Se analiza el performance de código y se buscan los cuellos de botella. Formato. IEEE 754 para representación de números racionales

#### **Sesión 3: Programación en Paralelo. (Sistemas de Memoria Compartida)**

En esta sesión se analiza el proceso de paralelización: descomposición, orquestación, asignación y mapeo. Se introducen los principios de un sistema coherente, consistente y se hace una descripción del hardware que permite lanzar varios hilos al mismo tiempo. Se empieza a trabajar la librería pthreads que permite la creación y manejo de hilos en C.

#### **Sesión 4: Programación en Paralelo. (Sistemas de Memoria Compartida)**

Se continúa trabajando con hilos. Se analiza que ocurre cuando dos o más hilos desean tener acceso a una posición de memoria al mismo tiempo, se analiza las variables de exclusión MUTEX como solución a este problema. Se trabaja la creación de hilos con atributos que les permitan sincronizarse con otros hilos y se generan barreras como punto de sincronismo entre ellos.

#### **Sesión 5: Programación en Paralelo. (Sistemas de Memoria Compartida)**

Se discute el soporte de hardware que permite que los hilos trabajen en paralelo, y las instrucciones de alto nivel que son usadas por el procesador. Se analiza la mejora en performance al lanzar varios hilos implementado semáforos y barreras reversas.

#### **Sesión 6: Repaso de FORTRAN**

En esta sesión se hará un repaso de todo el lenguaje FORTRAN, pasando por variables, condicionales, funciones, arreglos, cadenas, manejo del stack, el heap y archivos. Se analiza el performance de código y se buscan los cuellos de botella. Se analizan mejoras del compilador.

### **Sesión 7: Programación en Paralelo. (Sistemas Distribuidos)**

Se estudia como es que un programa puede ser distribuido en un clúster utilizando la interfaz MPI. Se detalla cómo se generan procesos dentro de un clúster y como se comunican entre ellos. Se analiza como la data se envía entre procesos de distintos nodos; variables, arreglos, tipos, etc.

### **Sesión 8: Programación en Paralelo. (Sistemas Distribuidos)**

Se analizan elementos de sincronización entre todos los procesos, semáforos y barreras. Para mejora de performance se ven funciones propias de MPI y reducciones y se compara el performance en comparación a envío de data tradicional.

### **Sesión 9: Programación en Paralelo. (Sistemas Distribuidos)**

Se analizan sistemas de archivo como medida para mejorar el performance distribuyendo un archivo entre todos los nodos de un clúster. Se trabaja con el formato hdf5 (.h5) estándar para archivos distribuidos en computación científica.

### **Sesión 10: Proyecto Final**

Como parte del curso, los alumnos deberán presentar un código que estén trabajando en el IGP paralelizado o bien usando Pthreads o MPI. El proyecto final debe como objetivo final mejorar el performance el programa mejorando el tiempo de ejecución y elevando el dataset que se analiza o trabaja, manteniendo la exactitud de este. Se utiliza LaTeX para la preparación del informe del proyecto final.