

Arquitecturas Paralelas

Taller 1

Introducción y Aplicaciones seriales

Contenido

- Prerequisitos
- Conceptos básicos
- Ambiente en *cluster*
- Ejercicios
- Problema (tarea)

Pre-requisitos

- Ambiente de programación
 - Lenguajes de programación
 - C/C++ (gcc, icc)
 - Fortran (g77, g90)
 - Build system (make, nmake, scons, ant, etc.)
 - Librerías de soporte (matemáticas, gráficas, etc.)
 - Editores ((x)emacs, vi(m), etc.)

Pre-requisitos ⁽²⁾

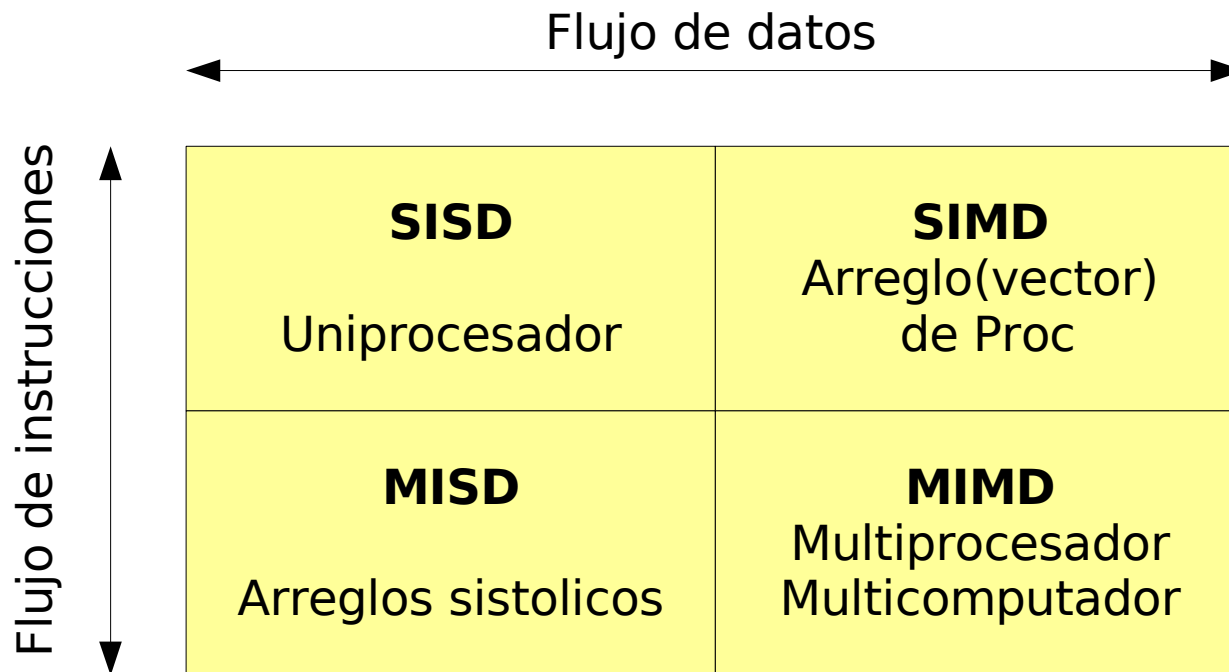
- Manejo de archivos y comandos del sistema
- Interprete de comandos (bash, tcsh, zsh, etc.)
- Acceso remoto
 - Remote Shell (rsh, rlogin)
 - Secure Shell (ssh, scp, sftp)

Pre-requisitos (3)

- Arquitectura H/W
 - Topología de interconexión
 - Almacenamiento de datos (particiones/discos)

Conceptos básicos

Taxonomía de Flynn



Conceptos básicos ⁽²⁾

- SMP
- Procesos Vs. Hilos
- Share Memory (Threads, OpenMP)
- Message Passing (MPI)
- Unificación entre los memoria compartida y paso de mensajes (upc)

Conceptos básicos ⁽³⁾

Clusters

- High Availability (HA)
- Load balancing
- High Performance (HPC)
 - Beowulf
 - COWs
 - Mosix
 - GRID
 - P2P

Ambiente en Cluster

- Arquitectura de software
 - Librería MPI (MPICH, LAM, OpenMPI)
 - Scheduler (PBS, Torque, Maui)
 - Monitoreo (Ganglia, OpenView)
 - Sistema de archivos (PVFS, NFS, Coda)
 - Debugger (TotalView, Jumpshot)
- Presentar el LiveCD DCC

Ejercicio 1 - Hola mundo!

Crear un archivo (*.c y *.h) en donde exista una función para recibir un nombre e imprimir un saludo

- saludo.h, saludo.c

- Crear un programa principal que pida un nombre y pase como parametro el nombre a la función definida previamente

- main.c

- Generar un Makefile para la compilación

Ejercicio 1 - Solución

```
/* saludo.h */  
  
void Saludo(char *nombre);
```

```
/* saludo.c */  
#include <stdio.h>  
  
void Saludo(char *nombre) {  
    printf("Hola %s!", nombre);  
}
```

Ejercicio 1 – Solución (2)

```
/* main.c */
#include <stdio.h>
#include "saludo.h"

int main(int argc, char *argv[]) {
    char nombre[30];
    printf("Nombre: ");
    scanf("%s", nombre);
    Saludo(nombre);
}
```

```
# Makefile

hola: main.c saludo.o
    gcc -o hola main.c saludo.o

saludo.o : saludo.h saludo.c
    gcc -c saludo.c
```

Ejercicio 2 - Multiplicación de matrices

- Generar un programa (usando módulos) para la multiplicación de matrices, donde se tenga una función que reciba 2 vectores y retorne un valor.

$$C = \sum_{i=1}^n A_i * B_i$$

- matriz.h, matriz.c
- main.c
- Uso de #define
- Makefile

Problema: Sistema de ecuaciones lineales

- Ecuaciones de la forma $A \cdot X = B$

$$\begin{aligned}4x_0 + 6x_1 + 2x_2 - 2x_3 &= 8 \\2x_0 + 5x_2 - 2x_3 &= 4 \\-4x_0 - 3x_1 - 5x_2 + 4x_3 &= 1 \\8x_0 + 18x_1 - 2x_2 + 3x_3 &= 40\end{aligned}$$

- Metodo de eliminación de Gauss: Busca obtener una matriz triangular superior, operando entre filas de la matriz aumentada

Problema: Sistema de ecuaciones lineales

- Tarea 1: Solución de ecuaciones por medio de eliminación de Gauss usando pivote parcial (solución serial)
 - Carga de matrices por archivo ($> 100 \times 100$)
 - Calculo de tiempo promedio de la solución
 - Gráfica de resultados $N/Tiempo$