

MPI

M. en C. José Alfredo Estrada Soto

¿Qué es?

- ▣ MPI (Interfaz de Paso de Mensajes) es un estándar para la implementación de sistemas de paso de mensajes desarrollado por un comité de proveedores, laboratorios del gobierno y universidades para funcionar en una amplia variedad de computadores paralelos y de forma tal que los códigos sean portables.

M. en C. José Alfredo Estrada Soto

¿Qué arquitectura requiere?

- ▣ Su diseño está inspirado en máquinas con una arquitectura de memoria distribuida,
 - en donde cada procesador es propietario de cierta memoria y la única forma de intercambiar información es a través de mensajes;
 - sin embargo, hoy en día también se encuentran implementaciones de MPI en máquinas de memoria compartida y redes de estaciones de trabajo.
- ▣ MPI representa un esfuerzo de la comunidad de programación paralela por estandarizar las subrutinas de comunicación en computadores de cómputo masivo.

M. en C. José Alfredo Estrada Soto

Implementación

- ▣ MPI no exige una determinada implementación del mismo.
- ▣ Lo importante es dar al programador una colección de funciones para que éste diseñe su aplicación, sin que tenga necesariamente que conocer el *hardware* concreto sobre el que se va a ejecutar, ni la forma en la que se han implementado las funciones que emplea.

M. en C. José Alfredo Estrada Soto

Enfoque práctico

- Imaginemos que queremos ejecutar el siguiente programa en 4 máquinas distintas de forma paralela:

```
int main(){  
    printf("Hola mundo\n");  
}
```

```
gcc -Wall hola.c -o hola  
./hola
```

- Una opción, es ir a las 4 máquinas, y ejecutar el programa en ellas (**POCO EFICIENTE**).

M. en C. José Alfredo Estrada Soto

- Otra opción es utilizar MPI, y desde una ~~maquina, ejecutar el siguiente código~~ (**MÁS EFICIENTE**)

```
Int main() {  
    MPI_Init(&argc,&argv);  
    printf("Hola mundo\n");  
    MPI_Finalize();  
}
```

```
mpicc -o hola hola.c  
mpirun -n 4 hola
```

M. en C. José Alfredo Estrada Soto

Proceso

- ▣ La *unidad básica* en MPI son los **procesos**.
 - Tienen espacios de memoria independientes.
 - Intercambio de información por paso de mensajes.
 - Introduce en concepto de comunicadores (grupo de procesos más contexto).
 - Cada proceso se le asigna un identificador interno propio de MPI (rank).

M. en C. José Alfredo Estrada Soto

Llamadas colectivas

- ▣ **MPI_Barrier()**
- ▣ **MPI_Bcast()**
- ▣ **MPI_Gather()**
- ▣ **MPI_Scatter()**
- ▣ **MPI_Alltoall()**
- ▣ **MPI_Reduce()**
- ▣ **MPI_Reduce_scatter()**
- ▣ **MPI_Scan()**

M. en C. José Alfredo Estrada Soto

Broadcast

- BROADCAST: envío de datos desde un proceso (root) a todos los demás.

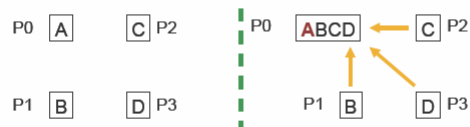


- `MPI_Bcast (&message, count, datatype, root, comm);`

M. en C. José Alfredo Estrada Soto

Gather

- GATHER: recolección de datos de todos los procesos en uno de ellos (orden estricto pid).

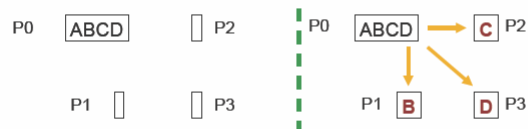


`MPI_Gather (&send_data, send_count,
send_type,&recv_data, recv_count,
recv_type,root, comm);`

M. en C. José Alfredo Estrada Soto

Scatter

- SCATTER: distribución de datos de un proceso entre todos los demás.

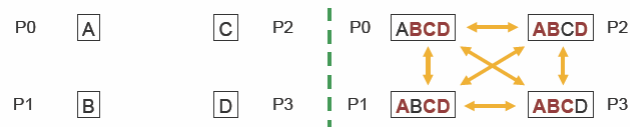


- `MPI_Scatter (&send_data, end_count, send_type, &recv_data, recv_count, recv_type, root, comm);`

M. en C. José Alfredo Estrada Soto

Allgather

- ALLGATHER: gather de todos a todos



- `MPI_Allgather(&send_data, send_count, send_type, &recv_buf, recv_count, recv_type, comm);`

M. en C. José Alfredo Estrada Soto

Reduce

- REDUCE: una operación de reducción con los datos de cada procesador, dejando el resultado en uno de ellos (root)



```
MPI_Reduce (&operand, &result, count,  
datatype,operator, root, comm);
```

M. en C. José Alfredo Estrada Soto

¿Qué herramientas existen?

- ▣ OpenMPI
 - Ubuntu
 - Debian
- ▣ Hay que agregar el entorno empleando el administrador de paquetes
- ▣ Verificar si se emplea MPI Mpich o Lam

M. en C. José Alfredo Estrada Soto