Algoritmos Distribuidos

Sistemas operativos

Gastón Aguilera

17 de noviembre de 2010

Menú del día

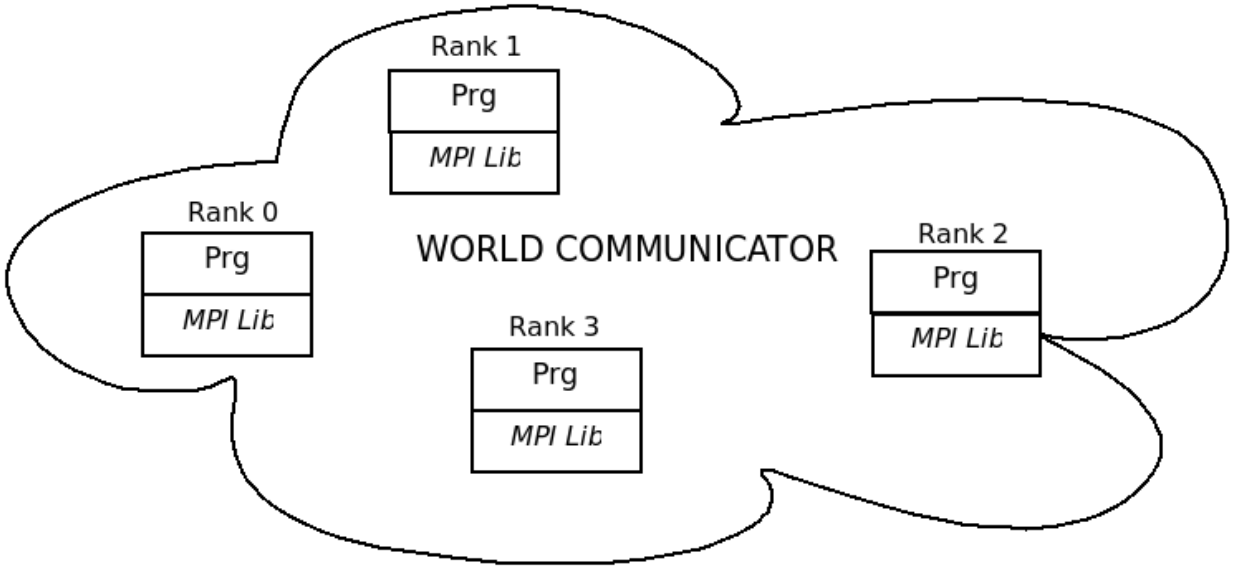
- Que es MPI?
- Ejecución de procesos distribuidos.
- Conceptos básicos de Communicators y Ranks.
- Funciones especiales para comunicadores *MPI_Split*, *MPI_Barrier*, *MPI_Bcast*.
- Comunicación entre procesos *MPI_Send*, *MPI_Recv*.
- TP3: Exclusion Mutua Distribuida.

MPI: *Message Passing Interface*

- Esta compuesto por un conjunto de funciones para intercambiar mensajes entre los procesos.
- MPI se basa en la ejecucion de un unico código ejecutable para todos los procesos.
- MPI v1: mpich, lam.
- MPI v2: openmpi, impi, lam (algunas funciones), mpich2.
- La version 2 de MPI tiene soporte para hilos (MPI_THREAD_SINGLE, MPI_THREAD_SERIALIZED).
- La ejecucion de procesos que utilizan MPI, se realiza usando: *mpirun, mpiexec* indicandosle la cantidad de procesos que se desean crear y la unica imagen ejecutable.
- Se utiliza el archivo .machines para indicar donde se ejecutaran los procesos.

Communicators & Ranks

- MPI utiliza rsh o ssh para ejecutar los procesos en las maquinas indicadas por .machines
- Todos los procesos tienen el mismo código, por lo tanto se deben poder diferenciar, de forma similar a la ejecucion de fork. En este caso cada proceso tiene un numero (RANK) asociado.
- El número de RANK está asociado a un COMMUNICATOR (MPI_COMM_WORLD).



Funciones iniciales MPI

- MPI_Init: Inicializa el entorno de MPI.
- MPI_Init: Finaliza el entorno de MPI.
- MPI_Com_Rank: Retorna el RANK del proceso que la invoca dentro del COMMUNICATOR indicado.
- MPI_Com_Size: Retorna la cantidad de procesos creados.

```
status= MPI_Init(&argc, &argv);
if (status!=MPI_SUCCESS)
{
   fprintf(stderr, "Error de MPI al inicializar.\n");
   MPI_Abort(MPI_COMM_WORLD, status);
}
MPI_Comm_size(MPI_COMM_WORLD, &np);
MPI_Comm_rank(MPI_COMM_WORLD, &rank);

MPI_Finalize();
```

Funciones MPI sobre COMMUNICATORS

- MPI_Split: Reagrupa a los procesos en nuevos COMMUNICATORS.
- MPI_Barrier: Espera a que todos los procesos lleguen a la barrera.
- MPI_Bcast: Envia mensaje a todos los procesos usando una estructura arborea de distribución.

```
// Reagrupa en dos comunicadores "iguales"
tipo=rank % 2;
MPI_Comm_split(MPI_COMM_WORLD, tipo, rank, &iguales);

MPI_Barrier(iguales);
MPI_Barrier(MPI_COMM_WORLD);

#define ROOT 0
MPI_Bcast(&variable, 1, MPI_INT, ROOT, MPI_COMM_WORLD);
```

Funciones MPI para manejo de MENSAJES

- MPI_Send: Envia un mensaje a un proceso particular en un communicator dado.
- MPI_Recv: Espera un mensaje de un proceso particular en un communicator dado.
- Hay tipos de mensajes predefinidos: MPI_INT, MPI_FLOAT, MPI_CHAR, MPI_BYTE, MPI_PACKED

```
MPI_Send(&variable, 1, MPI_INT, miserv, \\
            TAG_LIBERO, MPI_COMM_WORLD);

MPI_Recv(&variable, 1, MPI_INT, MPI_ANY_SOURCE, \\
            MPI_ANY_TAG, MPI_COMM_WORLD, &status);
```

Funciones MPI para manejo de MENSAJES

- Hay comodines MPI_ANY_SOURCE, MPI_ANY_TAG para usar en la funcion MPI_Recv.
- Para obtener información del sobre del mensaje recibido, se usa variable &status en el MPI_Recv.

```
MPI_Status status;
// status.MPI_SOURCE  status.MPI_TAG  status.MPI_ERROR

MPI_Recv(&variable, 1, MPI_INT, MPI_ANY_SOURCE, \\
            MPI_ANY_TAG, MPI_COMM_WORLD, &status);
origen=status.MPI_SOURCE;
tag=status.MPI_TAG;
```

Algoritmo de Exclusion Mutua Distribuida

Algoritmo de Exclusion Mutua Distribuida

- Se tiene N procesos MPI, con N=2*q. Generando así q procesos clientes (idcli 0 a q-1) (rank de clientes = 2 * r + 1)y q procesos servidores (idserv 0 a q-1) (rank de servidores = 2 * r).
- Los procesos clientes leen sus acciones de archivos, que tienen su idcli al final.
- Los servidores reciben mensajes de su cliente y de otros servidores segun el algoritmo del paper.
- En el paper se hace referencia a tres procesos para manejar la exclusion mutua, pero la implementación solo usa un servidor, que cumple la funcion de los tres procesos.
- Esta restricción hace que el servidor, sea implementado como una maquina de estados, que reacciona ante los pedidos de los servidores y del cliente.
- En el paper se hace mención a un numero de secuencia, y es la clave para resolver el ordenamiento de los pedidos para acceder a la sección critica.