Programación Paralela

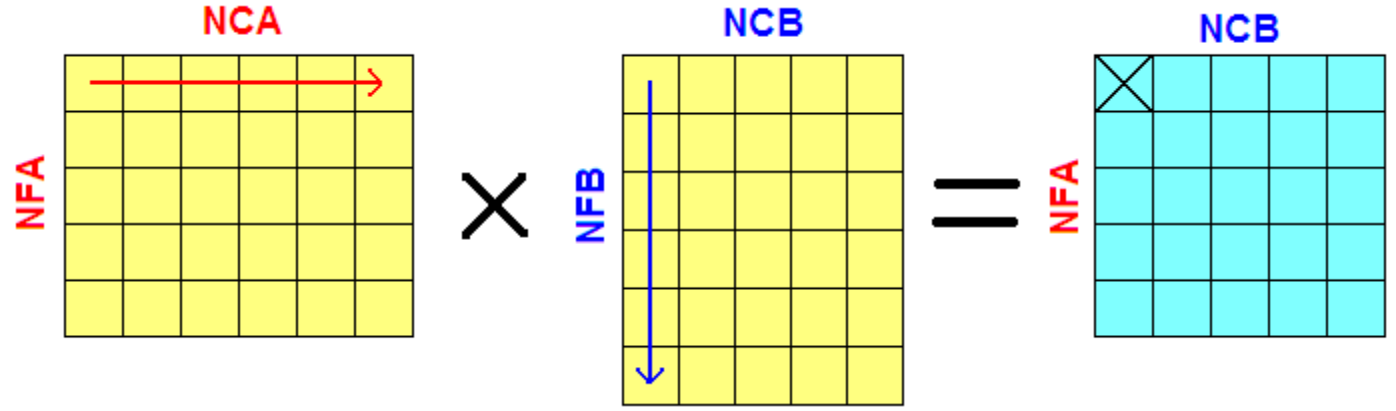
Multiplicación de Matrices
Grid 5000

Carmela Pozuelo

Multiplicación de Matrices
Esquema básico:

- Requisito para multiplicación:

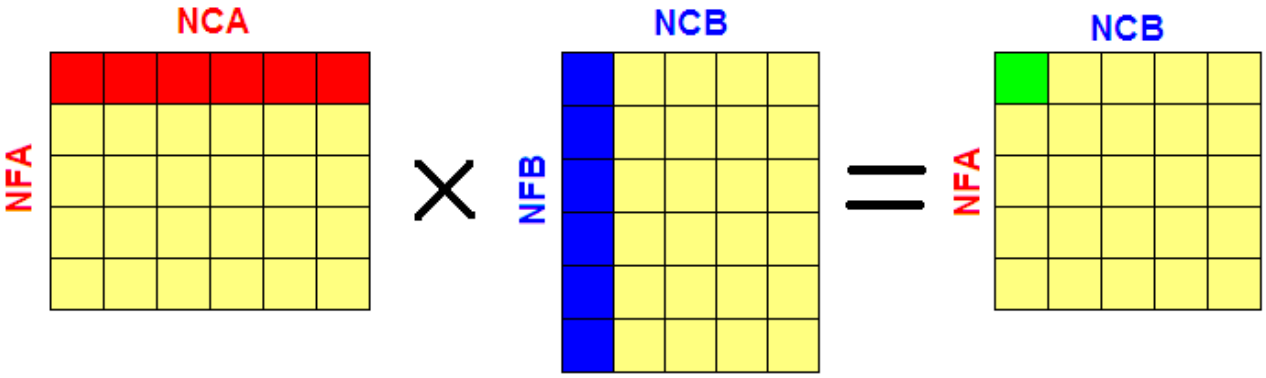
NCA = NFB



Multiplicación de Matrices
Procedimiento secuencial

- Un solo proceso que ejecuta:

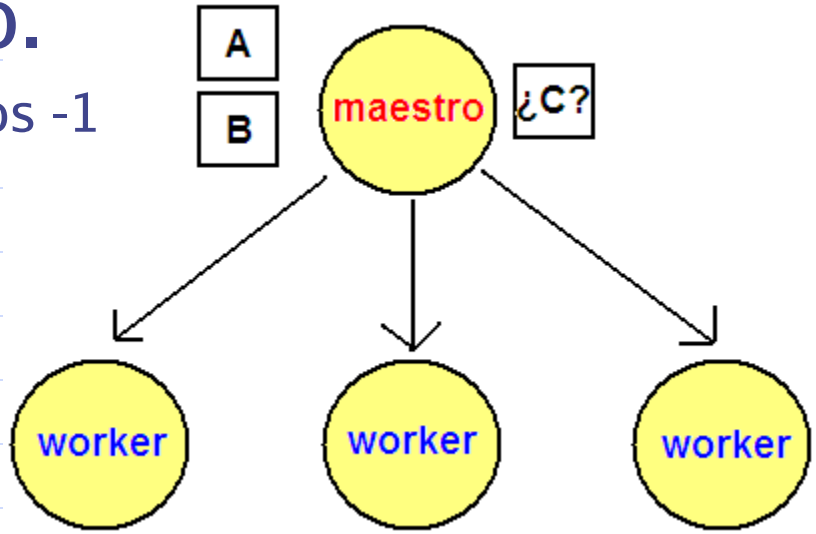
```
for(i=0; i<NFA; i++)
        for(j=0; j<NCB; j++) {
                c[i][j] = 0;
                for(k=0; k<NCA; k++)
                        c[i][j]=c[i][j]+a[i][k]*b[k][j];
        }
```



- Realizamos NFA*NCB*NCA iteraciones

Multiplicación de Matrices
Procedimiento paralelo: procesos

- Un proceso maestro → distribuye las distintas tareas al resto de procesos.
- Varios procesos worker → calcula un cierto número de filas de la matriz resultante y reenvía el resultado parcial al proceso maestro.
- Num_workers = num_procesos -1



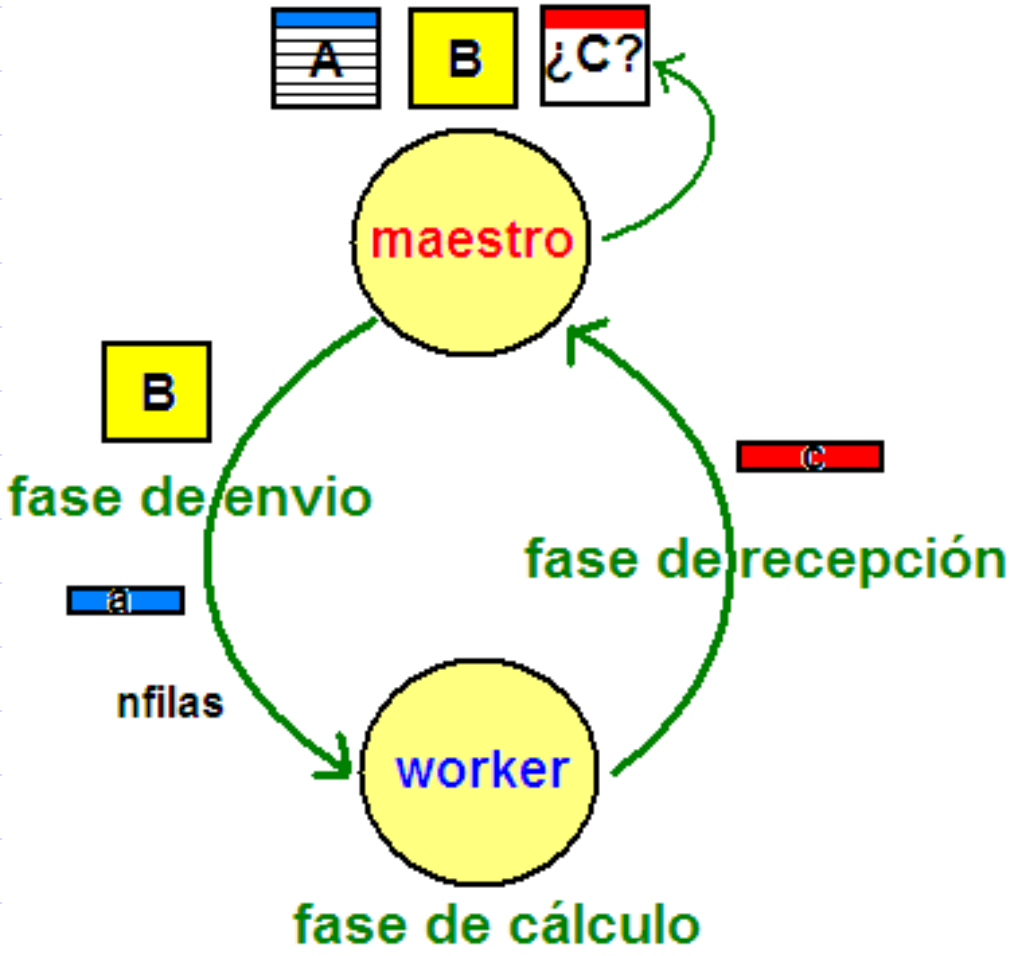
Multiplicación de Matrices
Procedimiento paralelo: proceso MAESTRO

- Conoce las matrices a multiplicar a y b
- Tareas a realizar por el MAESTRO:
  - Inicialización de las matrices “a” y “b”
  - Cálculo del número de filas que hay que enviar a cada proceso worker:
    - nfilas = NFA / num_workers
    - filas_extra = NFA % num_workers
  - ENVIO a cada WORKER:
    - Numero de filas de la matriz “a” → nfilas o nfilas+1
    - Elementos de “a” con los que debe operar → nfilas*NCA elementos de “a”
    - Matriz “b” en su totalidad.
  - ESPERA... (mientras los workers trabajan)

Multiplicación de Matrices
Procedimiento paralelo: proceso MAESTRO

- RECEPCIÓN de cada WORKER
  - Subparte de la matriz resultado → elementos de la "c" → nfilas*NCB elementos.
  - Atención!!! → ¿¿¿dónde metemos los elementos recibidos???
    - Respuesta: cada worker puede haber calculado un número de filas distintas, calcular cuántas y guardar resultado en matriz "c" con el desplazamiento adecuado.

Multiplicación de Matrices
Procedimiento paralelo: proceso MAESTRO



Fase de envío:

→ MPI_send

Fase recepción:

→ MPI_recv

Multiplicación de Matrices
Procedimiento paralelo: proceso WORKER

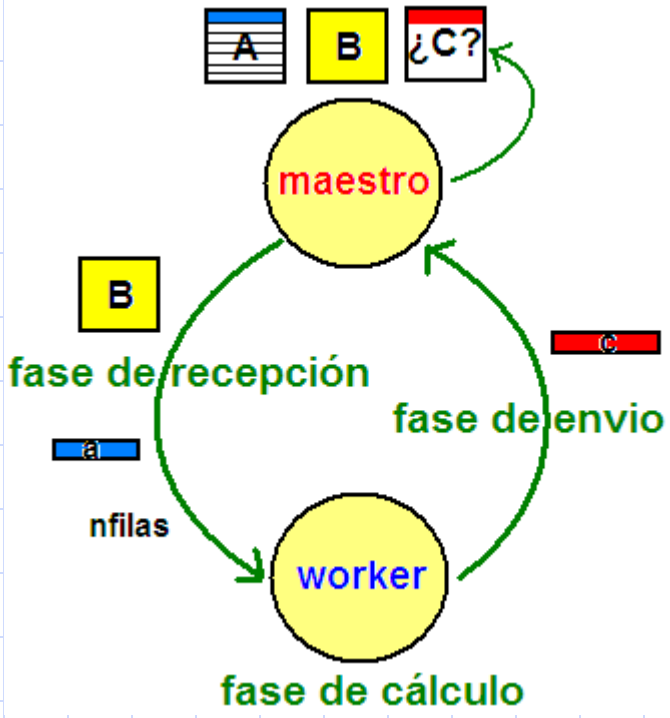
- Desconoce tanto la matriz a como la b
- Tareas a realizar por un WORKER:
  - Recepción del MAESTRO de:
    - Número de filas → nfilas
    - Elementos de "a" con los que va a calcular → nfilas*NCA elementos
    - Matriz "b" en su totalidad
  - Fase de Cálculo:

```
for(i=0; i<nfilas; i++)
   for(j=0; j<NCB; j++)
        c[i][j]=0; //inicializa la matriz
        for(k=0; k<NCA; k++)
             c[i][j] = c[i][j]+a[i][k]*b[k][j];
```

→Número de iteraciones = nfilas*NCA*NCB

Multiplicación de Matrices
Procedimiento paralelo: proceso WORKER

- Fase de ENVIO al MAESTRO:
  - Envío de la subparte de “c” calculada.



Fase de recepción

→MPI_recv

Fase de envío

→MPI_send

Multiplicación de Matrices
Procedimiento paralelo: programa común

- Todos los procesos ejecutan el mismo programa donde en el main tendremos:
  - if(taskid==0) //soy el maestro
    - inicializo a y b
    - envio trabajo a los workers
    - recibo resultado de workers
  - if(taskid>0) //soy un worker
    - recibo trabajo del maestro
    - realizo cálculos
    - envío resultado al maestro

Multiplicación de Matrices
Procedimiento paralelo: directivas MPI

- ENVIO MAESTRO → WORKERS
- Bucle for(i=0;i<num_workers;i++)
  - Envio del numero de filas de "a"
    - Calculamos nfilas para el worker numero i

MPI_send(&nfilas,1,MPI_INT,**i**,FROM_MASTER,MPI_COMM_WORLD)

  - Envio de nfilas de la matriz "a"

MPI_send(a[**offset**],nfilas*NCA,MPI_DOUBLE,i,FROM_MASTER,MPI_COMM_WORLD)

  - Envio de la matriz "b" completa

MPI_send(b[0],NFB*NCB,MPI_DOUBLE,i,FROM_MASTER,MPI_COMM_WORLD)

Multiplicación de Matrices
Procedimiento paralelo: directivas MPI

- RECEPCIÓN WORKERS ← MAESTRO
  - Recepción del numero de filas de “a”

```
MPI_recv(&nfilas,1,MPI_INT,MASTER,FROM_MASTER,MPI_COMM_WORLD,&status)
```

  - Recepción de nfilas de la matriz “a”

```
MPI_recv(a,nfilas*NCA,MPI_DOUBLE,0,FROM_MASTER,MPI_COMM_WORLD,&status)
```

  - Recepción de la matriz “b” completa

```
MPI_recv(b,NFB*NCB,MPI_DOUBLE,0,FROM_MASTER,MPI_COMM_WORLD,&status)
```

- ENVIO de resultado WORKER → MAESTRO
  - Tras calcular la submatriz de c correspondiente

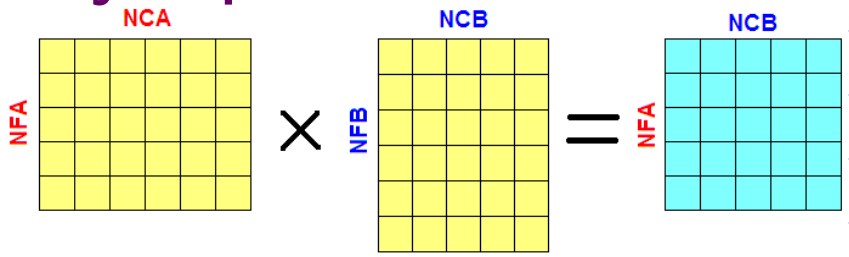
```
MPI_send(c,nfilas*NCB,MPI_DOUBLE,0,FROM_WORKER,MPI_COMM_WORLD)
```

- RECEPCION resultado MAESTRO ← WORKER
  - En un bucle for(i=0; i<numworkers; i++)
    - Calculo nfilas que voy a recibir

```
MPI_recv(c[offset],nfilas*NCB,MPI_DOUBLE,i,FROM_WORKER,MPI_COMM_WORLD,&status)
```

    - Actualizo offset

Multiplicación de Matrices
Procedimiento paralelo: ejemplo



- NCA=6 NFA=5
- NCB=5 NFB=6
- Ejecución secuencial: NFA*NCB*NCA= 5*5*6 = 150 iteraciones
- Ejecución paralela con 2 workers = nfilas*NCB*NCA
  - Primer worker→ nfilas=3 →3*5*6=90 iteraciones
  - Segundo worker→ nfilas=2 →2*5*6=60iteraciones
  - 1'6 veces más rápido que en secuencial
- Ejecución paralela con 5 workers
  - Cada worker → nfilas=1 → 1*5*6=30 iteraciones
  - 5 veces más rápido que en secuencial

Grid'5000

Plataforma de experimentación

Francia

Introducción
Grid y Grid Computing

- GRID: infraestructura que permite la integración y el uso colectivo de ordenadores de alto rendimiento, redes y bases de datos que son propiedad y están administrados por diferentes instituciones.
- GRID COMPUTING: tecnología innovadora que permite utilizar de forma coordinada todo tipo de recursos que no están sujetos a un control centralizado.
  - Nueva forma de computación distribuída
  - Recursos heterogéneos

Grid'5000
Introducción

- Plataforma de experimentación
- 10 laboratorios franceses implicados
- ¿Para qué?: construir una plataforma que permita a los desarrolladores de la comunidad (registrados), validar los distintos niveles de software creados para la puesta en marcha de las tecnologías grid → grid computing.
- Intenta conseguir:
  - rapidez de calculo y capacidad de almacenamiento
  - Utilización de red jerárquica de máquinas
  - Permitir al usuario de introducir sus aplicaciones

Grid'5000
Introducción

- Los clusters que componen la grille sólo son accesibles desde los otros clusters de la grille.
- acces.site.grid5000.fr : los puntos de acceso a grid5000, a partir de los cuales podemos conectarnos desde el exterior para acceder a todos sus recursos.
- OAR: herramienta que nos permite reservar las máquinas que vamos a utilizar. La reserva es un requisito para trabajar en grid5000, si no reservas un nodo, no puedes utilizarlo.
  - Oar.site.grid5000.fr
- Podemos crear una imagen del entorno que nos interese para nuestro experimento.
- Podemos desarrollar nuestra imagen en las máquinas que hemos reservado para facilitar nuestras experimentos utilizando la herramienta KADEPLOY.
- Sincronización de los distintos clusters utilizando la herramienta RSYNC.

Grid'5000
Creación de una imagen

- Nos conectamos a grid a través de la máquina de acceso de nancy:

  ssh carmela@acces.nancy.grid5000
- Nos conectamos a la máquina que nos permite hacer las reservas:

  ssh oar.nancy.grid5000.fr
- Reservamos una máquina

  oarsub –I –q deploy (1 máquina por defecto)
- Desarrollamos una imagen por defecto

  Kdeploy –e fedora4all –m máquina –p sda6
- Realizamos las modificaciones que queramos
- La comprimimos y la registramos

  tar –czlf mi_imagen.tgz –numeric-owner /

  karecordenv –n mi_imagen –fb ruta/mi_imagen.tgz ...

Grid'5000
Utilización de nuestra imagen

- Hacemos la reserva de N máquinas con la herramienta OAR:

  ```
  >> oarsub -I -q deploy -l nodes=N
  ```

- Fichero $OAR_FILE_NODES : contiene los nombres de todas las máquinas reservadas
- Desarrollamos nuestra imagen en las máquinas reservadas ($OAR_FILE_NODES) con KADEPLOY:

```
>> kadeploy -e mi_imagen -f $OAR_FILE_NODES
   -p sda6
```

- Máquinas disponibles con nuestra imagen!!!

Grid'5000
Estado de las máquinas en Toulouse

Grid5000 Toulouse OAR nodes

Summary:

OAR node status	Free	Busy	Total
Nodes	12	28	57

Reservations:

node-1.toulouse.grid5000.fr	67443	node-2.toulouse.grid5000.fr	67438	node-3.toulouse.grid5000.fr	67443	node-4.toulouse.grid5000.fr	Free	node-5.toulouse.grid5000.fr	Suspected
node-6.toulouse.grid5000.fr	Free	node-7.toulouse.grid5000.fr	Suspected	node-8.toulouse.grid5000.fr	67442	node-9.toulouse.grid5000.fr	Free	node-10.toulouse.grid5000.fr	Free
node-11.toulouse.grid5000.fr	67443	node-12.toulouse.grid5000.fr	67438	node-13.toulouse.grid5000.fr	67434	node-14.toulouse.grid5000.fr	Suspected	node-15.toulouse.grid5000.fr	Suspected
node-16.toulouse.grid5000.fr	Suspected	node-17.toulouse.grid5000.fr	67438	node-18.toulouse.grid5000.fr	Down	node-19.toulouse.grid5000.fr	Suspected	node-20.toulouse.grid5000.fr	67442
node-21.toulouse.grid5000.fr	67434	node-22.toulouse.grid5000.fr	67443	node-23.toulouse.grid5000.fr	67434	node-24.toulouse.grid5000.fr	Free	node-25.toulouse.grid5000.fr	Free
node-26.toulouse.grid5000.fr	Suspected	node-27.toulouse.grid5000.fr	67438	node-28.toulouse.grid5000.fr	Free	node-29.toulouse.grid5000.fr	Suspected	node-30.toulouse.grid5000.fr	67438
node-31.toulouse.grid5000.fr	67442	node-32.toulouse.grid5000.fr	Free	node-33.toulouse.grid5000.fr	67442	node-34.toulouse.grid5000.fr	Suspected	node-35.toulouse.grid5000.fr	Free
node-36.toulouse.grid5000.fr	Suspected	node-37.toulouse.grid5000.fr	67443	node-38.toulouse.grid5000.fr	Free	node-39.toulouse.grid5000.fr	67443	node-40.toulouse.grid5000.fr	Free
node-41.toulouse.grid5000.fr	67442	node-42.toulouse.grid5000.fr	67443	node-43.toulouse.grid5000.fr	67438	node-44.toulouse.grid5000.fr	67434	node-45.toulouse.grid5000.fr	Free
node-46.toulouse.grid5000.fr	67442	node-47.toulouse.grid5000.fr	Suspected	node-48.toulouse.grid5000.fr	Suspected	node-49.toulouse.grid5000.fr	67443	node-50.toulouse.grid5000.fr	Suspected
node-51.toulouse.grid5000.fr	67438	node-52.toulouse.grid5000.fr	67443	node-53.toulouse.grid5000.fr	Suspected	node-54.toulouse.grid5000.fr	67438	node-55.toulouse.grid5000.fr	67443
node-56.toulouse.grid5000.fr	Suspected	node-57.toulouse.grid5000.fr	Down						

Grid'5000
Detalle de las reservas en Toulouse

Job details:

Id	User	State	Queue	NbNodes	Weight	Type	Properties	Reservation	Walltime	Submission Time	Start Time	Scheduled Start
67201	hbouziane	Waiting	deploy	50	1	PASSIVE	p.deploy = "YES"	Scheduled	15:30:00	2007-01-16 14:06:14	2007-01-21 10:30:00	2007-01-21 10:30:00
67311	ejeanvoine	Waiting	deploy	50	1	PASSIVE	p.deploy = "YES"	Scheduled	11:00:00	2007-01-17 11:13:36	2007-01-22 21:00:00	2007-01-22 21:00:00
67313	ejeanvoine	Waiting	deploy	50	1	PASSIVE	p.deploy = "YES"	Scheduled	11:00:00	2007-01-17 11:23:48	2007-01-23 21:00:00	2007-01-23 21:00:00
67417	deploy	Waiting	deploy	51	1	PASSIVE	p.deploy = "YES"	Scheduled	17:00:00	2007-01-19 17:52:26	2007-01-22 02:50:00	2007-01-22 02:50:00
67434	ccoufort	Running	default	4	1	PASSIVE		None	37:00:00	2007-01-19 20:47:11	2007-01-19 20:47:13	2007-01-19 20:47:13
67438	cmijoule	Running	default	8	1	PASSIVE		None	29:00:00	2007-01-20 04:45:37	2007-01-20 04:45:38	2007-01-20 04:45:38
67442	cdenis	Running	deploy	6	1	INTERACTIVE	p.deploy = "YES"	None	12:00:00	2007-01-20 13:48:32	2007-01-20 13:48:33	2007-01-20 13:48:33
67443	ldorazio	Running	default	10	1	PASSIVE		Scheduled	11:59:58	2007-01-20 16:05:05	2007-01-20 16:05:07	2007-01-20 16:05:07

Grid'5000
Caso real de utilización

- Una nueva plataforma para el cálculo distribuido → HiPoP
- ¿Es escalable? → test en grid'500
- Imagen creada:
  - Plataforma
  - Usuario
  - Ssh
  - Scripts
  - …