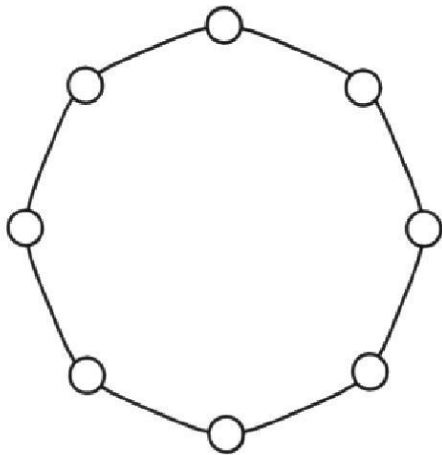


# Message Passing Interface MPI

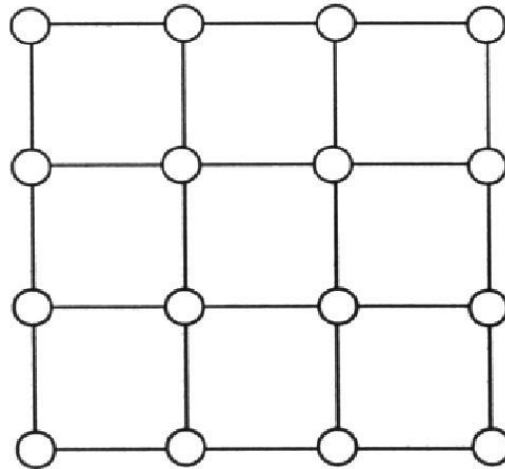
Dalle lezioni di Calcolo Parallelo  
del Prof. A. Murli

## Le topologie

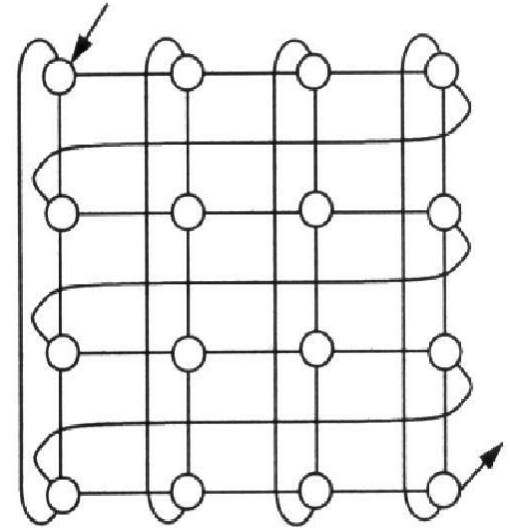
# Esempi di topologie



Anello



Griglia



Toro

L'utilizzo di una topologia per la progettazione di un algoritmo in ambiente MIMD è spesso **legata** alla geometria "intrinseca" del problema in esame.

# Definizione: *topologia*

Una topologia è  
**la geometria "virtuale"**  
in cui si immaginano disposti  
i processi.



La topologia "virtuale" in cui sono disposti  
i processori può **non avere alcun nesso**  
con la disposizione "reale" dei processori!

Griglia

```
#include <stdio.h>
#include "mpi.h"
/* Scopo: definizione di una topologia
    a griglia bidimensionale nproc=row*col */
main(int argc, char **argv)
{ int menum,nproc,row,col;
  int dim,*ndim,reorder,*period,*coordinate;
  MPI_Comm comm_grid;

  MPI_Init(&argc,&argv);
  MPI_Comm_rank(MPI_COMM_WORLD,&menum);
  MPI_Comm_size(MPI_COMM_WORLD,&nproc);
/* Numero di righe della griglia di processo */
  if (menum == 0)
  {   printf("Numero di righe della griglia");
      scanf("%d",&row); }
/* Spedizione di row da parte di 0 a tutti i processi */
  MPI_Bcast(&row,1,MPI_INT,0,MPI_COMM_WORLD);
/* Definizione del numero di colonne della griglia */
  col = nproc/row;
/* Numero di dimensioni della griglia */
  dim = 2;
  coordinate = (int*)calloc(dim,sizeof(int));
```

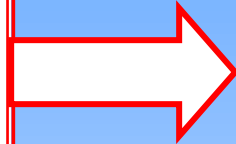
```
/* vettore contenente le lunghezze di ciascuna dimensione*/
ndim = (int*)calloc(dim,sizeof(int));
ndim[0] = row;
ndim[1] = col;
/* vettore contenente la periodicità delle dimensioni */
period = (int*)calloc(dim,sizeof(int));
period[0] = period[1] = 0;
reorder = 0;
/* Definizione della griglia bidimensionale */

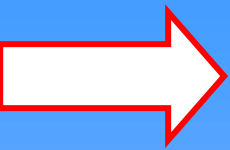
MPI_Cart_create(MPI_COMM_WORLD,dim,ndim,period,reorder,
                &comm_grid);

MPI_Comm_rank(comm_grid,&menum_grid);
/* Definizione delle coordinate di ciascun processo
   nella griglia bidimensionale */

MPI_Cart_coords(comm_grid,menum,dim,coordinate);

/* Stampa delle coordinate */
printf("Processore %d coordinate nella griglia
       (%d,%d) \n",menum,*coordinate,* (coordinate+1));
MPI_Finalize();
return 0; }
```

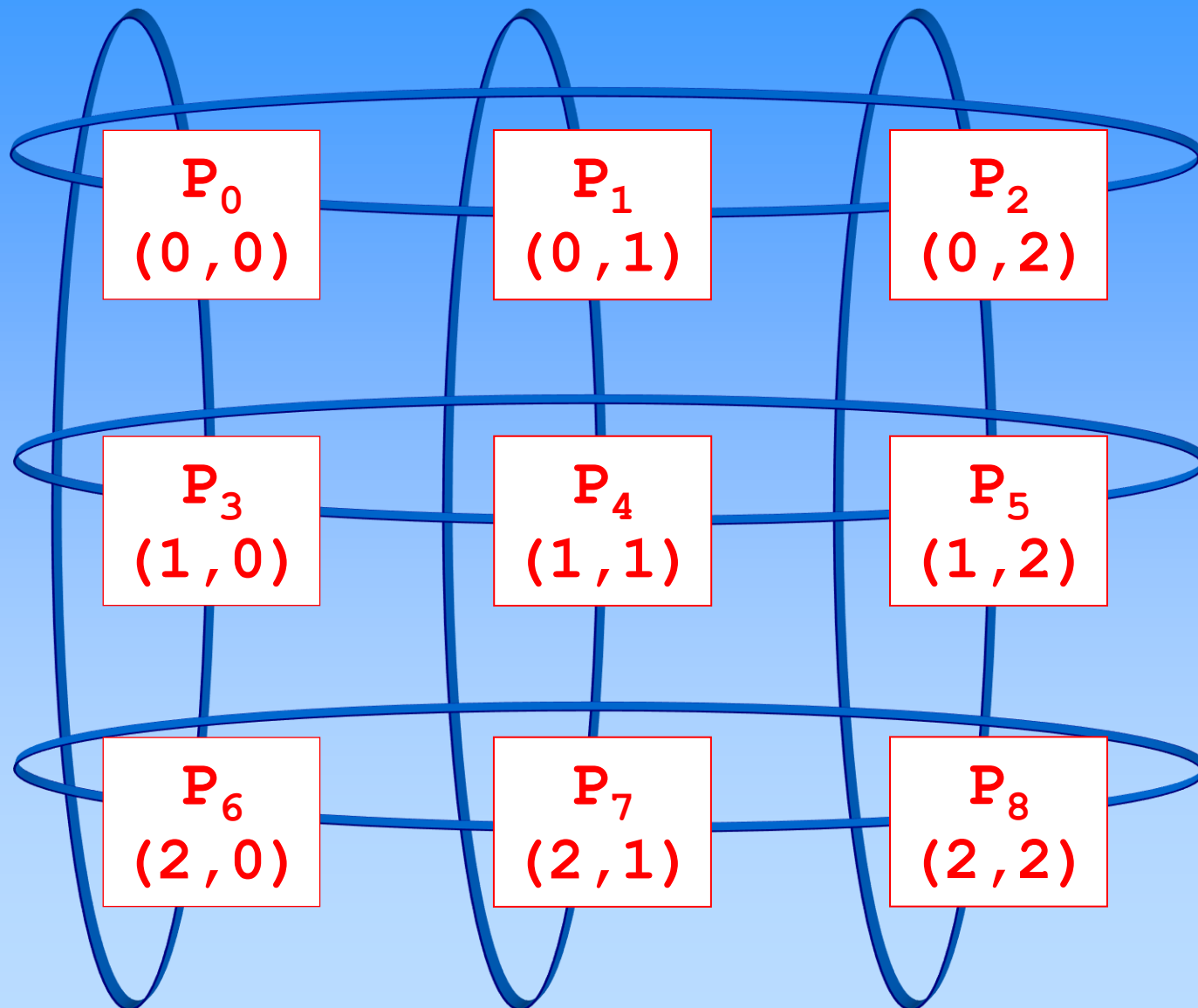




```
MPI_Cart_create(MPI_COMM_WORLD,dim,ndim,period,  
reorder,&comm_grid);
```

- Ogni processo dell'ambiente `MPI_COMM_WORLD` definisce la griglia denominata `comm_grid`, di dimensione 2 (`dim`) e non periodica lungo le due componenti (`period[i]=0, i=0,1`). Il numero di righe e di colonne della griglia sono memorizzati rispettivamente nella prima e nella seconda componente del vettore `ndim`. I processi non sono riordinati secondo un particolare schema (`reorder=0`).





```
MPI_Cart_create(MPI_Comm comm_old, int dim,  
               int *ndim, int *period,  
               int reorder,  
               MPI_Comm *new_comm) ;
```

- Operazione collettiva che restituisce un nuovo communicator **new\_comm** in cui i processi sono organizzati in una griglia di dimensioni **dim**.
- L'*i*-esima dimensione ha lunghezza **ndim[i]**.
- Se **period[i]=1**, la *i*-esima dimensione della griglia è periodica; non lo è se **period[i]=0**.

```
MPI_Cart_create(MPI_Comm comm_old, int dim,  
                int *ndim, int *period,  
                int reorder,  
                MPI_Comm *new_comm) ;
```

**comm\_old**    communicator di input

**dim**        numero di dimensioni della griglia

**\*ndim**       vettore di dimensione **dim** contenente  
le lunghezze di ciascuna dimensione

**\*period**     vettore di dimensione **dim** contenente la  
periodicità di ciascuna dimensione

**reorder**     permesso di riordinare i **menum** (1=si; 0=no)

**\*new\_comm**   communicator di output associato alla griglia

```
/* vettore contenente le lunghezze di ciascuna dimensione */
ndim = (int*)calloc(dim,sizeof(int));
ndim[0] = row;
ndim[1] = col;

/* vettore contenente la periodicità delle dimensioni */
period = (int*)calloc(dim,sizeof(int));
period[0] = period[1] = 0;
reorder = 0;

/* Definizione della griglia bidimensionale */

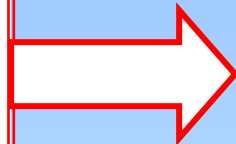
MPI_Cart_Create(MPI_COMM_WORLD,dim,ndim,period,reorder,
                &comm_grid);

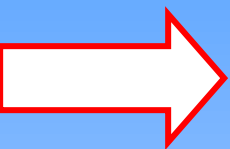
MPI_Comm_rank(comm_grid,&menum_grid);

/* Definizione delle coordinate di ciascun processo
   nella griglia bidimensionale */

MPI_Cart_coords(comm_grid,menum,dim,coordinate);

/* Stampa delle coordinate */
printf("Processore %d coordinate nella griglia
       (%d,%d) \n",menum,*coordinate,* (coordinate+1));
MPI_Finalize();
return 0; }
```





```
MPI_Cart_coords(comm_grid, menum, dim, coordinate);
```

- Ogni processo **menum** calcola le proprie 2 (**dim**) coordinate (**coordinate<sub>i</sub>**,  $i=0,1$ ) nell'ambiente **comm\_grid**.

```
MPI_Cart_coords(MPI_Comm comm_grid,  
                 int menum_grid, int dim,  
                 int *coordinate);
```

- Operazione collettiva che restituisce a ciascun processo di **comm\_grid** con identificativo **menum\_grid**, le sue coordinate all'interno della griglia predefinita.
- **coordinate** è un vettore di dimensione **dim**, i cui elementi rappresentano le coordinate del processo all'interno della griglia.

**Fine Esercitazione**