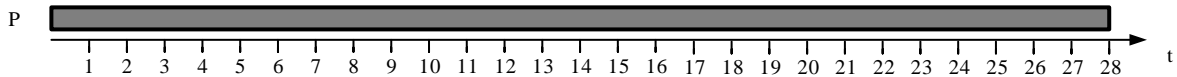
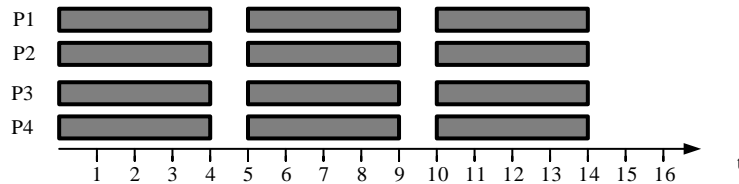


Soru 1. Aşağıda Cannon matris çarpma algoritmasının ana döngüsü verilmiştir. Bu döngüyü iletişim ve işlem çakışacak biçimde yeniden kodlayınız.

```
/* Get into the main computation loop */  
for (i=0; i<dims[0]; i++) {  
    MatrixMultiply(nlocal, a, b, c); /*c=c+a*b*/  
  
    /* Shift matrix a left by one */  
    MPI_Sendrecv_replace(a, nlocal*nlocal, MPI_DOUBLE,  
        leftrank, 1, rightrank, 1, comm_2d, &status);  
  
    /* Shift matrix b up by one */  
    MPI_Sendrecv_replace(b, nlocal*nlocal, MPI_DOUBLE,  
        uprank, 1, downrank, 1, comm_2d, &status);  
}
```

Soru 2. Soru 1 kapsamında hızlanma (Speedup) ve etkinlik (Efficiency) ölçütlerinin, iletişimle işlemin çakıştırılmasından nasıl etkilendiğini kısaca açıklayınız.

Soru 3. Aşağıdaki grafiklerden hızlanma (Speedup) ve etkinlik (Efficiency) değerlerini hesaplayınız. İlk grafik dört işlemcili bir sistemdeki uygulamanın çalışmasını, alttaki grafik de uygulamanın en iyi seri algoritma ile gerçekleştirmesini göstermektedir.



Kullanabileceğiniz MPI işlemlerinden bazıları:

```
int MPI_Init(int *argc, char ***argv)

int MPI_Finalize()

int MPI_Comm_size(MPI_Comm comm, int *size)

int MPI_Comm_rank(MPI_Comm comm, int *rank)

int MPI_Send(void *buf, int count, MPI_Datatype datatype, int dest, int
             tag, MPI_Comm comm)

int MPI_Recv(void *buf, int count, MPI_Datatype datatype, int source, int
            tag, MPI_Comm comm, MPI_Status *status)

int MPI_Isend(void *buf, int count, MPI_Datatype datatype,
             int dest, int tag, MPI_Comm comm, MPI_Request *request)

int MPI_Irecv(void *buf, int count, MPI_Datatype datatype,
            int source, int tag, MPI_Comm comm, MPI_Request *request)

int MPI_Test(MPI_Request *request, int *flag, MPI_Status *status)

int MPI_Wait(MPI_Request *request, MPI_Status *status)

int MPI_Sendrecv(void *sendbuf, int sendcount, MPI_Datatype senddatatype,
                int dest, int sendtag, void *recvbuf, int recvcount,
                MPI_Datatype recvdatatype, int source, int recvtag,
                MPI_Comm comm, MPI_Status *status)

int MPI_Get_count(MPI_Status *status, MPI_Datatype datatype, int *count)

int MPI_Cart_create(MPI_Comm comm_old, int ndims, int *dims, int
                  *periods, int reorder, MPI_Comm *comm_cart)

int MPI_Cart_rank(MPI_Comm comm_cart, int *coords, int *rank)

int MPI_Cart_coord(MPI_Comm comm_cart, int rank, int maxdims, int
                  *coords)

int MPI_Cart_shift(MPI_Comm comm_cart, int dir, int s_step, int
                  *rank_source, int *rank_dest)
```

Kullanabileceğiniz değişmezler: `MPI_COMM_WORLD`, `MPI_ANY_SOURCE`, `MPI_ANY_TAG`

Yanıt 1.

Yanıt 2.

Yanıt 3.