

Problem 2 gözüme daha kolay görüldüğü için ondan başlamak istiyorum.

2 matris var.

$$\begin{array}{c|c} a_1 & a_2 \\ \hline a_3 & a_4 \end{array}$$

$$\begin{array}{c|c} b_1 & b_2 \\ \hline b_3 & b_4 \end{array}$$

$$\begin{array}{c|c} s_1 & s_2 \\ \hline s_3 & s_4 \end{array}$$

$$s_1 = (a_1 \cdot b_1) + (a_2 \cdot b_3)$$

Matrislerimin
2 boyutlu

$$s_2 = (a_1 \cdot b_2) + (a_2 \cdot b_4)$$

bu yüzden

$$s_3 = (a_3 \cdot b_1) + (a_4 \cdot b_3)$$

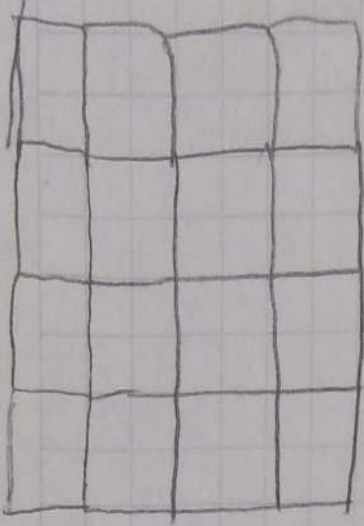
iç içe 2for

ile çalışmalıyız.

$$s_4 = (a_3 \cdot b_2) + (a_4 \cdot b_4)$$

Sonuç matrisini oluştururken her bir sütun veya satırı komple toplayıp oluşturduğumuz için 1 for a daha ihtiyacımız var.

Bir sonuç matrisindeki hücrede dururken bizim için diğer matrislerde gezicek.



4x4 lük matris üzerinden
düşünelim.

buna sonuç matrisi olarak
düşünüp çekirdeklere parçalar
oluşturmalarmı söylesek

4 çekirdek çalışıyor olsa

her bir satırı 1 çekirdeğe hesaplatıp
en sonunda birleştirebilirim.

peki satır oluşturmak için neye ihtiyacım
var?

A matrisinden
sadece
çekirdeğin oluşturacağı
satırı alsak yeterli.

B matrisinden
tüm değerler
gerekliyor $\frac{n}{4}$

yapılacak işlemleri yazarsak

A matrisini işlemci sayısına böl

Parçaları işlemcilere dağıt

B matrisini işlemcilere ver

çarma işlemini yap

işlemcilerin oluşturduğu ufak matrisleri

Sırasıyla al

ve sonuç matrisinde birleştir.

$[0, 0, 0]$, $\rightarrow A_1 \times b$ herş

$[0, 0, 0]$, $\rightarrow A_2 \times b$ herş

$[0, 0, 0]$ $\rightarrow A_3 \times b$ herş

Satır bunları çekirdeklerle hesaplatıp
ana matrisi oluşturmak için birleştiriyoruz.

Bu işlemin seri kodunu yazalım

```
double a[2][2];
```

```
double b[2][2];
```

```
double c[2][2];
```

```
double toplam = 0;
```

```
for (int sat = 0; sat < 2; sat++) {
```

```
    for (int sut = 0; sut < 2; sut++) {
```

```
        for (int gez = 0; gez < 2; gez++) {
```

```
            Toplam += a[sat][gez] * b[gez][sut];
```

```
        }
```

```
        c[sat][sut] = toplam;
```

```
        toplam = 0;
```

```
    }
```

```
}
```

Peki bunu nasıl paralel hale getiririz?

en dıştaki satırları döşeyecek ve içinde en çok işlem barındıran 'for' un çalışma sayısını azaltırız.

$A[100][100], B[100][100], C[100][100]$

MPI init

MPI comm rank = b

MPI comm size = a

init arrays with 1 and 2

offset = 0

for i in a:

~~MPI_send~~ give offset to i.core

give $A[\text{offset}]$ to i.core

give B to i.core

offset++;

for i in a:

~~MPI_recv~~ take all send variables

Multiply A with B to C

for i in a:

~~MPI_send~~ send c to 0.core

if b = 0:

for i in a:

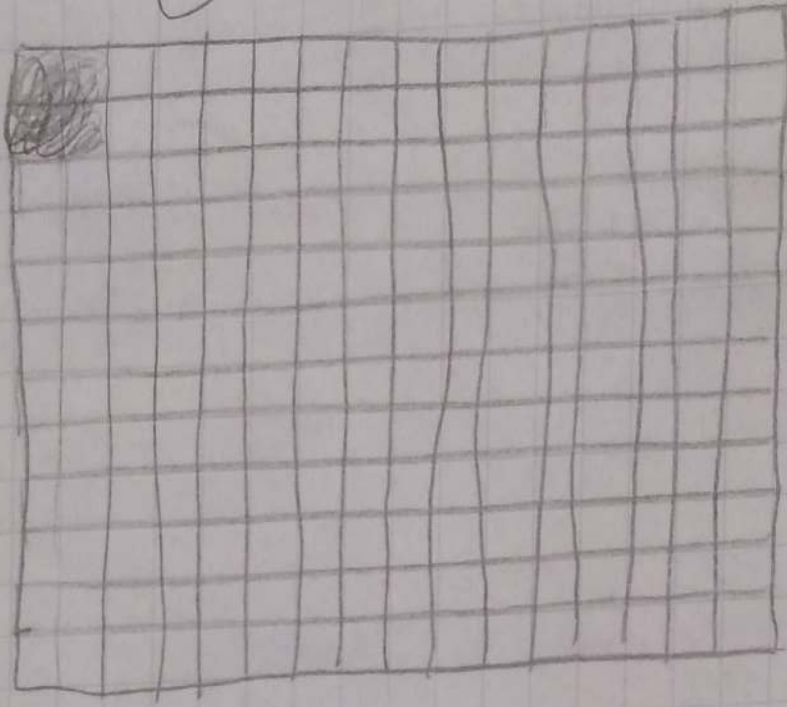
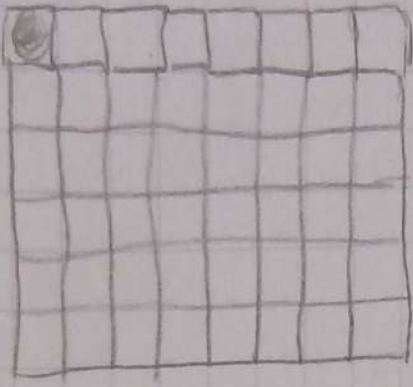
receive all send c to core offset

print(c)

5

Problem 1

8



Her bir piksel 4 katına
çizmiş yani elimizde
matrisin satırlarını
geçirdelerimize

yollarsak

geçirdelerimize
da

bunları

2ye katlayıp

geri

yollarsa

bu iş olur.

Satır sayısını işlemci sayısına böl

her bir işlemciye kaç satır düşüyorsa böl ve yolla

her bir elemanın yanına kopyasını koy

satırı, kople duplicate et

0'a yolla

geçirdiğin geldiğin hizaya göre sonucu matrisin
koy

sonucu baş

Problem 1

$R = 600$; $col = 800$

mpi-init

mpi_comm_rank

Mpi_comm_size

matrix[R][col];

$sbr = R/15$;

hiza = 0;

tempmatrix[sbr*2][col*2];

sonmatrix[R*2][col*2];

for i in is :

MPI_send(&hiza, 1, MPI_INT, i, 1, MPI_COMM_WORLD)

hiza++

dikayyer = hiza*sbr;

yatayyer = 0;

MPI_recv(&hiza)

for (i = 0; i < sbr*2; i = i+2)

for (j = 0; j < col*2; j = j+2)

eleman = matrix[dikayyer][yatayyer];

tempmatrix[i][j] = eleman // sol üst

tempmatrix[i+1][j+1] = eleman // sağ alt

tempmatrix[i][j+1] = eleman // sağ üst

tempmatrix[i+1][j] = eleman // sol alt

yatayyer++; dikayyer++;

sonmatrix.append(tempmatrix[i], tempmatrix[i+1])

temp oluşturu sıra da yollayıp birleştir-
mekte

for i in is:

mpi_send(&hiza, 1, mpi_int, i, 1, mpi_comm_world);

mpi_send(&sonmatris[hiza][0], (64*2)*(64*2),
mpi_double, i, 1, mpi_comm_world);

if id = 0:

Print(sonmatris) //bilgiyorum c de yok
ama kaba kod
neticede 285

for i in is:

mpi_recv(&hiza, 1, mpi_int, i, 2, mpi_comm_world)

mpi_recv(&sonmatris[hiza][0], (64*2)*(64*2),

mpi_double, i, 1, mpi_comm_world);