

16.430 normal

*

```
int main()
```

```
float A[N], B[N], C[N], num;
```

```
int i;
```

```
num = 0;
```

```
for (i = 0; i < N; i++) {
```

```
    C[i] = A[i] + B[i];
```

```
    num = num + C[i];
```

```
}
```

```
return 0;
```

```
float sum[N];
```

```
RANK, p (comm = MPI_COMM_WORLD
```

```
{ /* puts a punto */
```

```
{ N, A[N], B[N], C[N], sum }
```

```
if (RANK == 0) {
```

```
    for (i = 0; i < p; i++) {
```

```
        MPI_SEND(&N, 1, MPI_INT, i, 0, comm);
```

```
    }
```

```
    for (i = 0; i < p; i++) {
```

```
        MPI_SEND(&A[i * N/p], N/p, MPI_FLOAT, i, 0, comm);
```

```
        MPI_SEND(&B[i * N/p], N/p, MPI_FLOAT, i, 0, comm);
```

```
    }
```

```
    for (i = 0; i < N/p; i++) {
```

```
        C[i] = A[i] + B[i];
```

```
        sum[i] = sum + C[i];
```

```
    }
```

```
    for (i = 0; i < p; i++) {
```

```
        MPI_RECV(&C[i * N/p], N/p, MPI_FLOAT, i, 0, comm, &status);
```

```
    }
```

do {

MPI_Recv(&N, 1, MPI_INT, 0, 0, comm, &status);

float A[N/P], B[N/P], C[N/P], sum;

MPI_Recv(A, N/P, MPI_float, 0, 0, comm, &status);

MPI_Recv(B, N/P, MPI_float, 0, 0, comm, &status);

for (i=0; i < N/P; i++) {

C[i] = A[i] + B[i];

}

MPI_Send(C, N/P, MPI_float, 0, 0, comm);

...

}

for (i=0; i < N; i++) {

sum = sum + C[i];

}

// Sumamos todos los valores de sum.

Está claro que esto podría ser más optimo

amaris

objetivo

MPZ - Broadcast ($N, 1, \text{MPZ_INT}, 0, \text{comm}$): // Ya comentado anteriormente

float $A[\frac{N}{p}]$, $B[\frac{N}{p}]$, $C[\frac{N}{p}]$, $\text{sum}[\frac{N}{p}]$;

MPZ - Scatter ($A, \frac{N}{p}, \text{MPZ_Float}, A_c, \frac{N}{p}, \text{MPZ_Float}, 0, \text{comm}$);

MPZ - Scatter ($B, \frac{N}{p}, \text{MPZ_Float}, B_c, \frac{N}{p}, \text{MPZ_Float}, 0, \text{comm}$);

for ($i=0; i < \frac{N}{p}; i++$) {

$C[i] = A_c[i] + B_c[i]$

MPZ - Gather ($C, \frac{N}{p}, \text{MPZ_Float}, C_c, \frac{N}{p}, \text{MPZ_Float}, 0, \text{comm}$);

for ($i=0; i < N; i++$) {

$\text{sum} = C[i] + \text{sum}$;

// esto se puede mejorar.

}