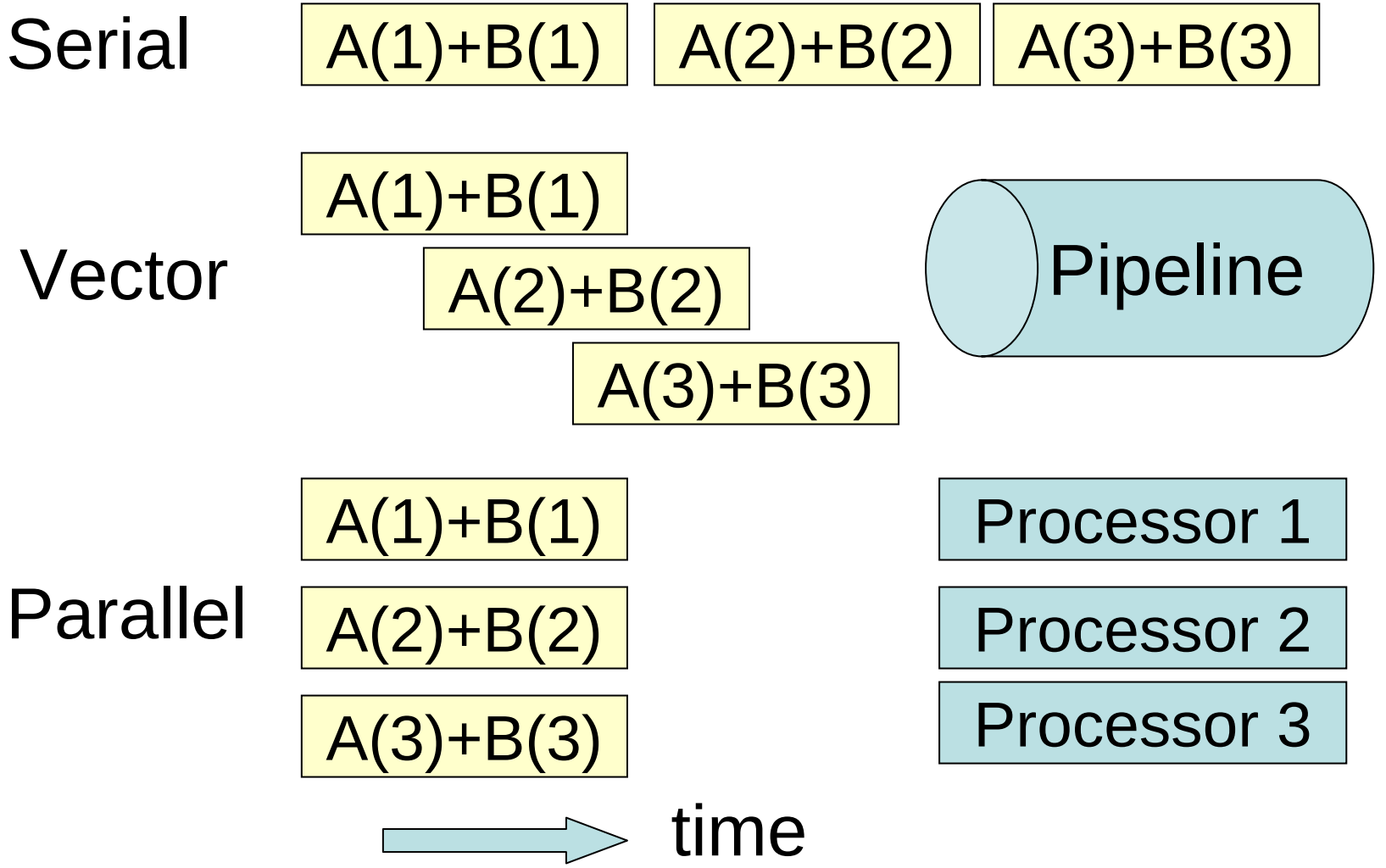
Introduction to vector/parallel computers

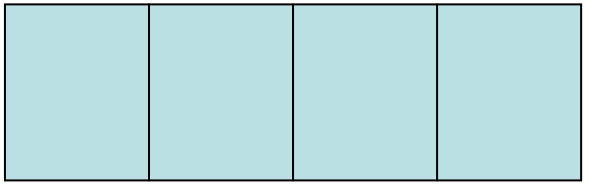
Ryoji Matsumoto
(Chiba University)

High Performance Computers



What's Pipeline ?

Pipeline for A+B

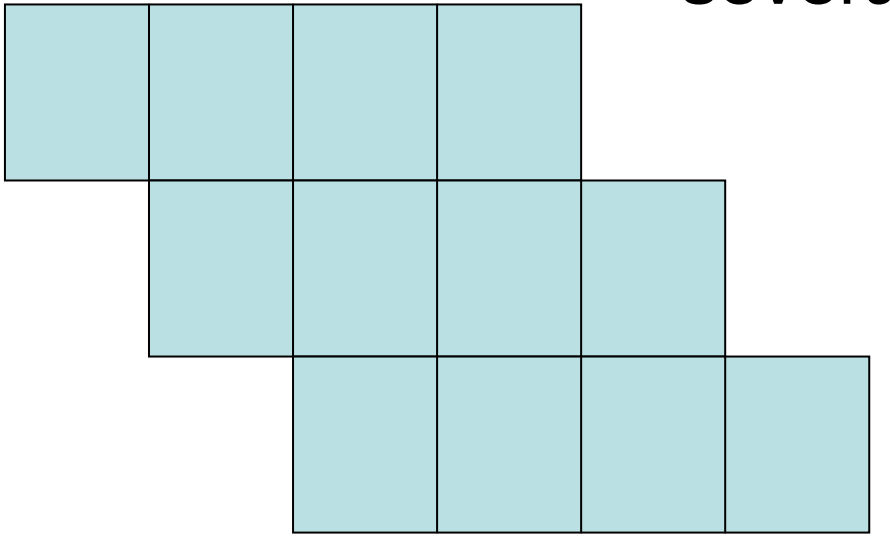


Operation can be divided into several steps

A(1)+B(1)

A(2)+B(2)

A(3)+B(3)

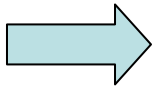


Some Hints to Get High Performance in Vector Processors

- Use long vector

```
do  i  = 1, 10
C(i)=A(i)+B(i)
enddo
```

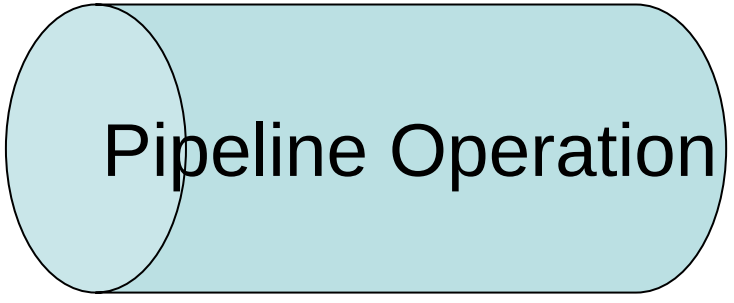
slower



```
do  i  = 1, 100
C(i)=A(i)+B(i)
enddo
```

faster

Initial stage



Final stage

Some Hints to Get High Performance in Vector Processors

- Avoid operations which wait the completion of previous operation

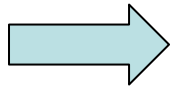
A(i+1)=A(i-1)+A(i)

A(3)=A(1)+A(2)

A(4)=A(2)+A(3)

- Access memory sequentially

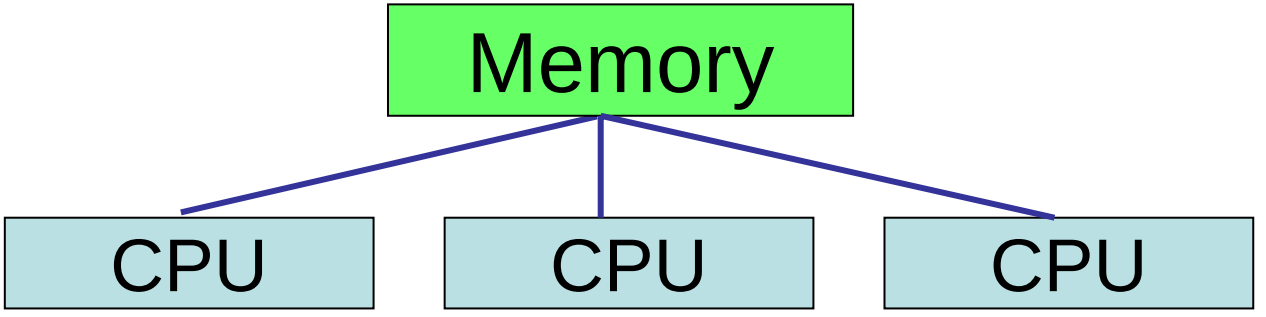
```
do  i = 1, ix
do  j = 1, jx
a(i,j)=b(i,j)+c(i,j)
enddo
enddo
```



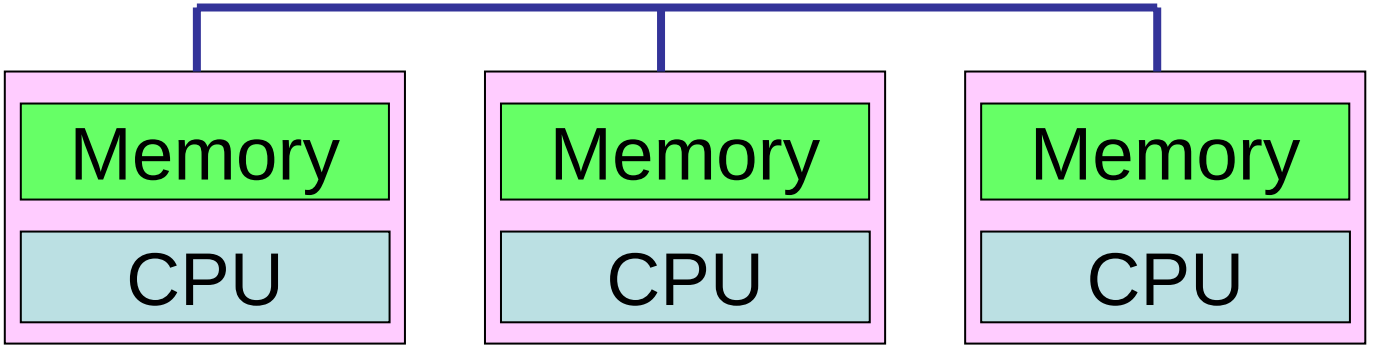
```
do  j = 1, jx
do  i = 1, ix
a(i,j)=b(i,j)+c(i,j)
enddo
enddo
```

Parallel Computers

- Shared Memory Machines

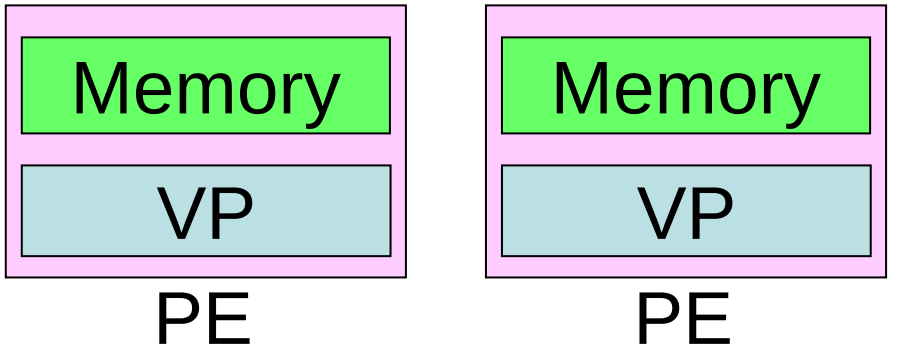


- Distributed Memory Machines



Vector/Parallel Computers

- VPP5000 @ NAOJ

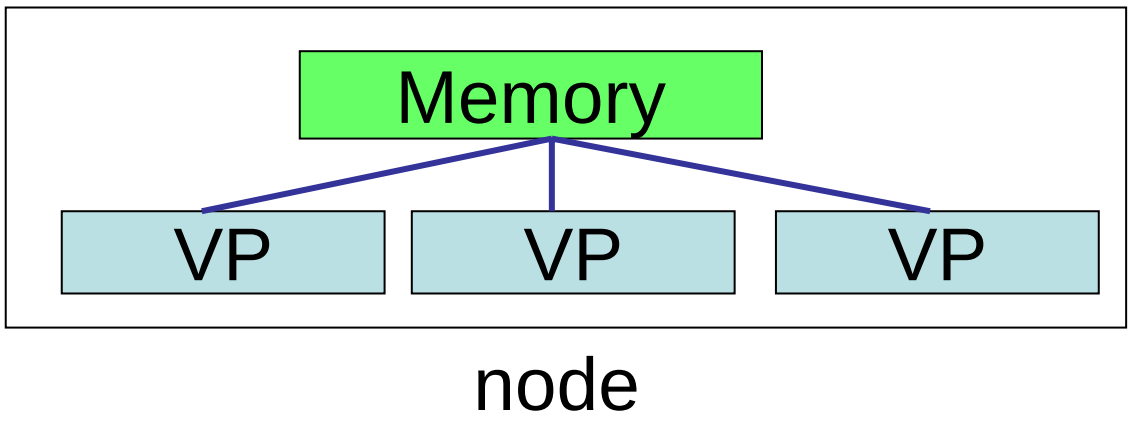


60PES

8GFLOPS/PE

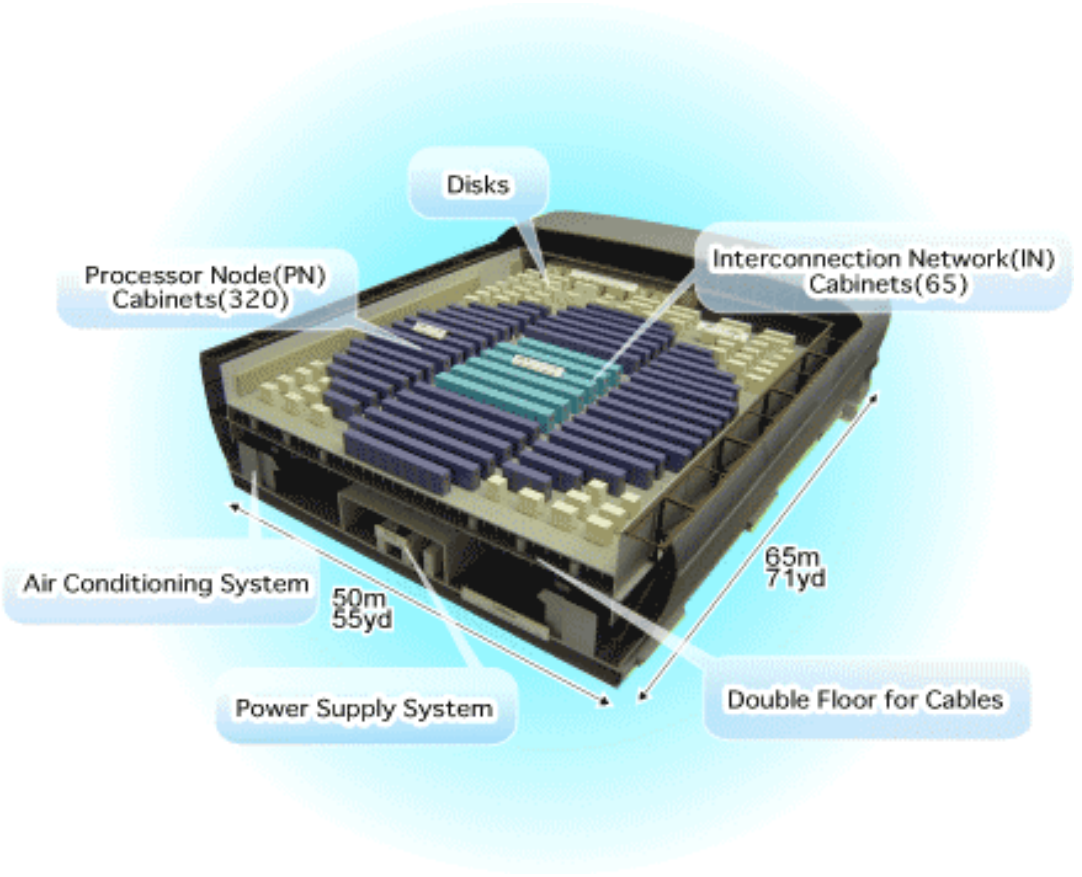
16GB memory/PE

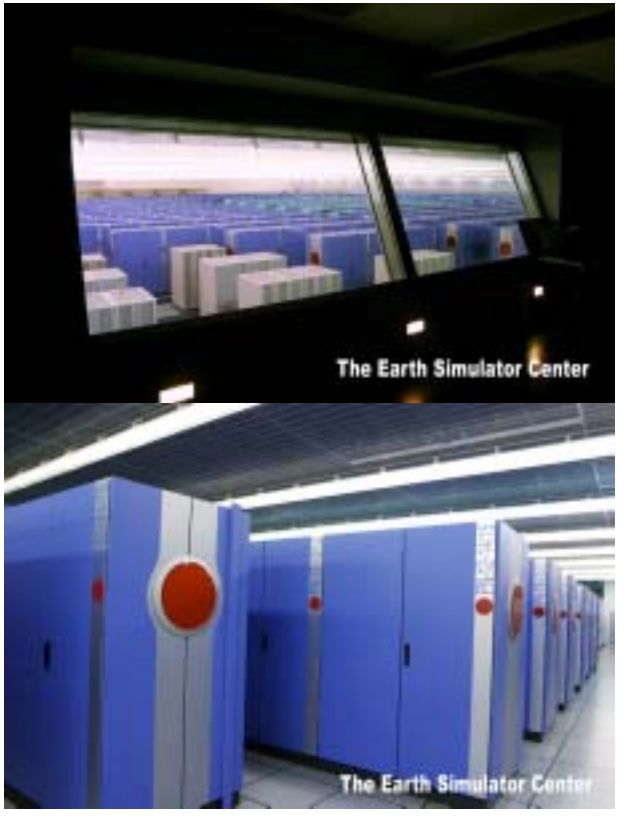
- Earth Simulator @jamstec



640 nodes
8 processors in each node
8GFLOPS/PE
16GBmemory/ node

Earth Simulator





640 nodes * 8 = 5120 processors , 40TFLOPS

16GB memory in each node

How to Parallelize a Code

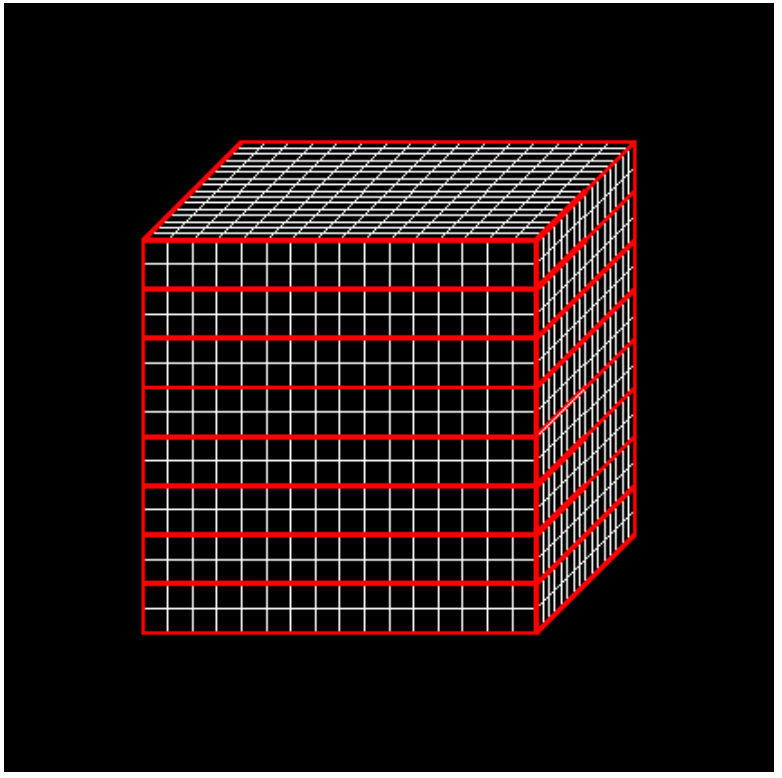
- Shared Memory Machines
  - autoparallel
- Distributed Memory Machines
  - Message Passing

    MPI
  - Data parallel language

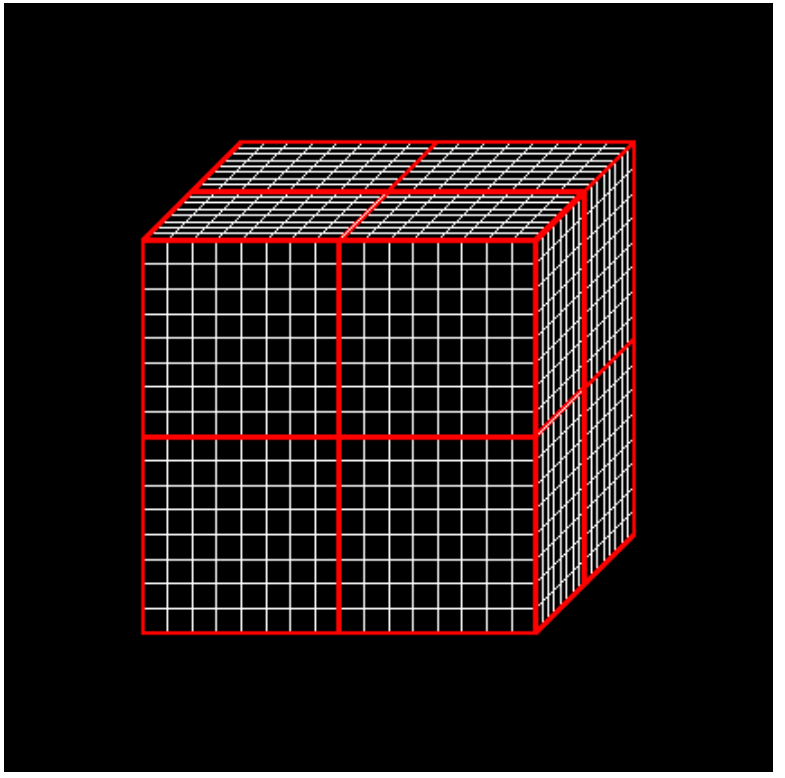
    HPF, VPP-FORTRAN

Parallelization of MHD Codes

- Domain Decomposition

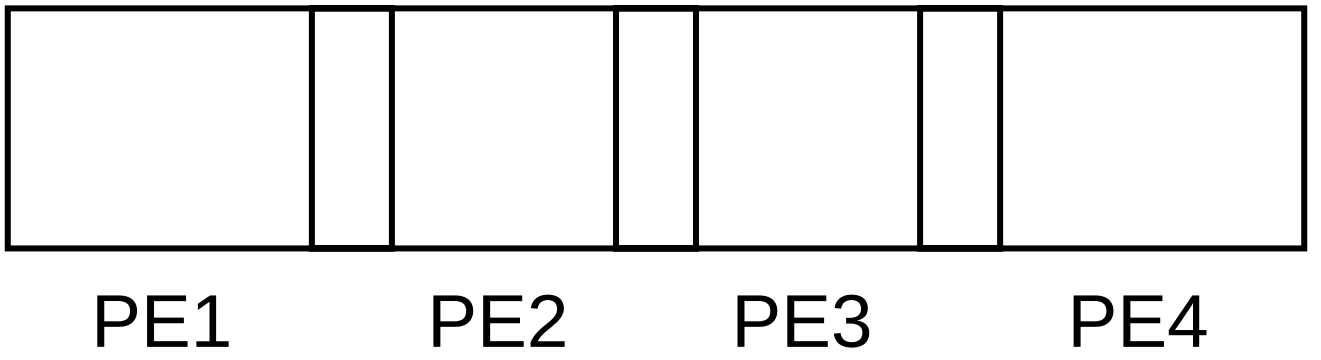


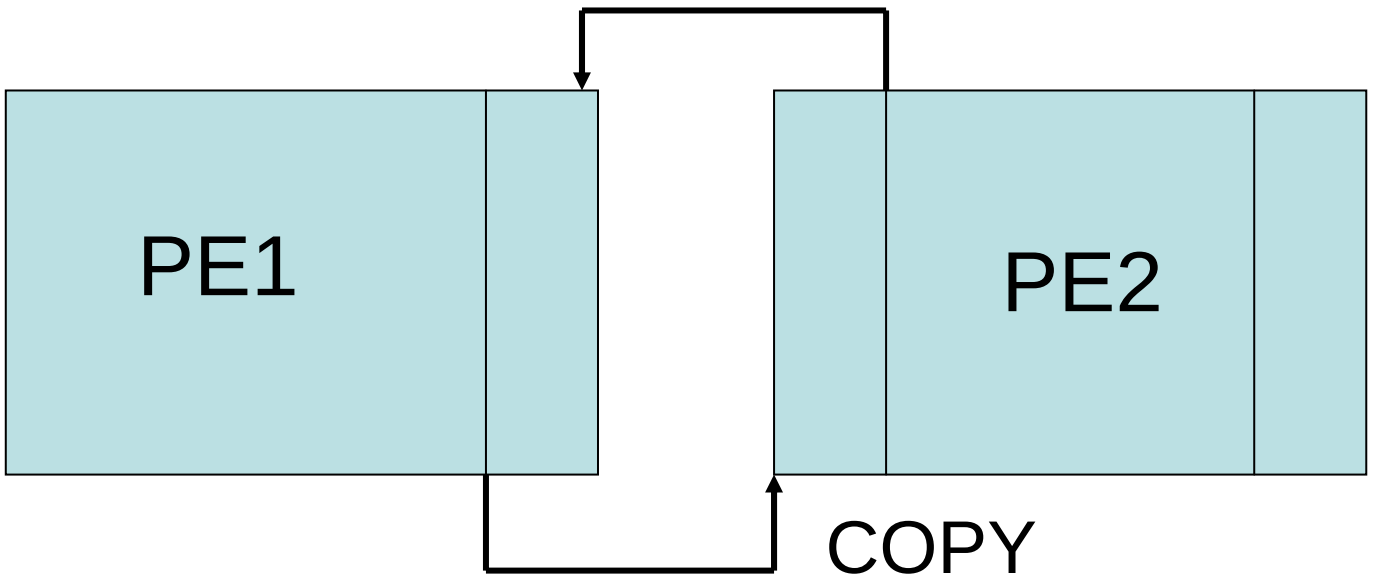
1D



3D

Communication between PEs in explicit MHD codes





Parallelized Modules in CANS

Please check modules with name mdp_....

They are parallelized by using MPI

Data-exchange modules are in the directory “commonmpi”