Profiling – III and Revision

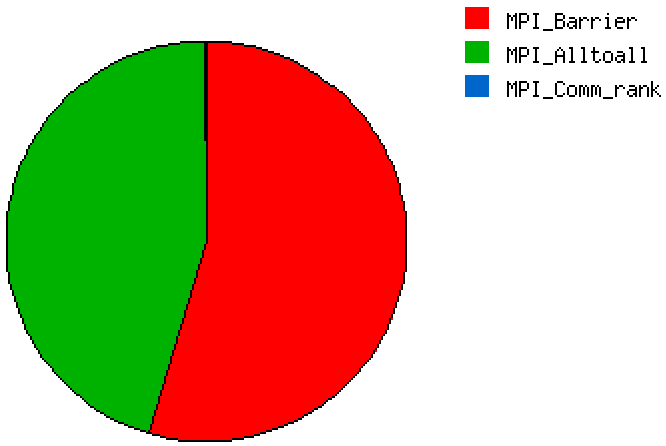
Lecture 25

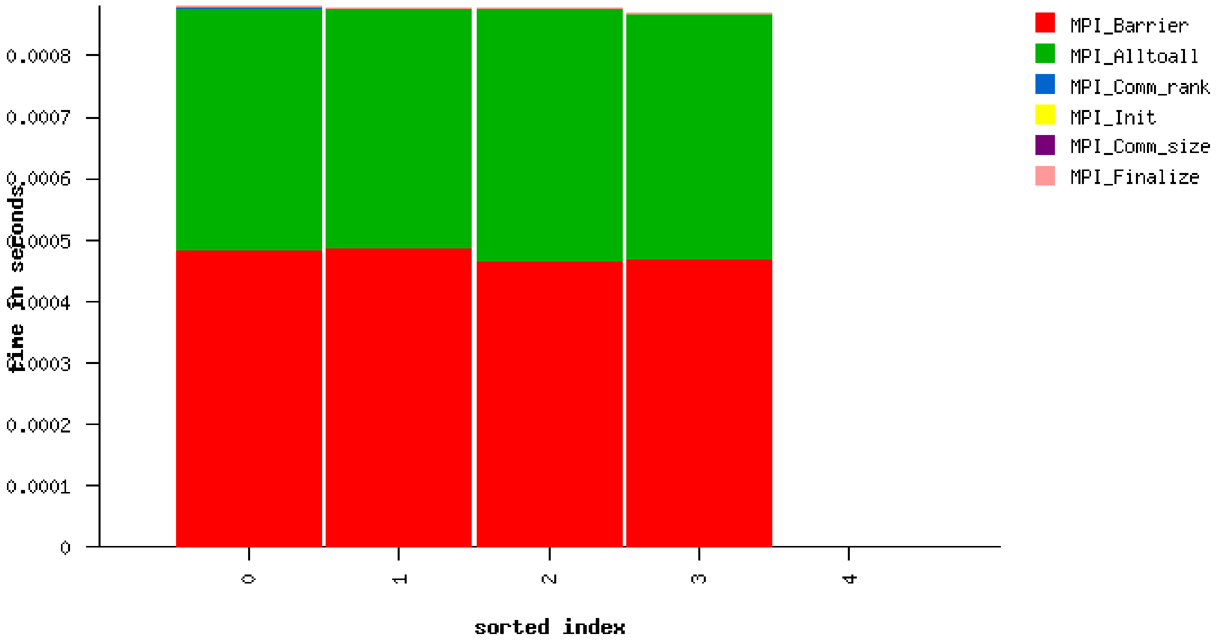
April 17, 2024

Profiling

```
for (int i=0; i<50; i++)
 {
   MPI_Barrier (MPI_COMM_WORLD);
   MPI_Alltoall(message, arrSize, MPI_INT, recvMessage, arrSize, MPI_INT, MPI_COMM_WORLD);
 }
```

Alltoall - NPROCS=4, Data size = 4 KB

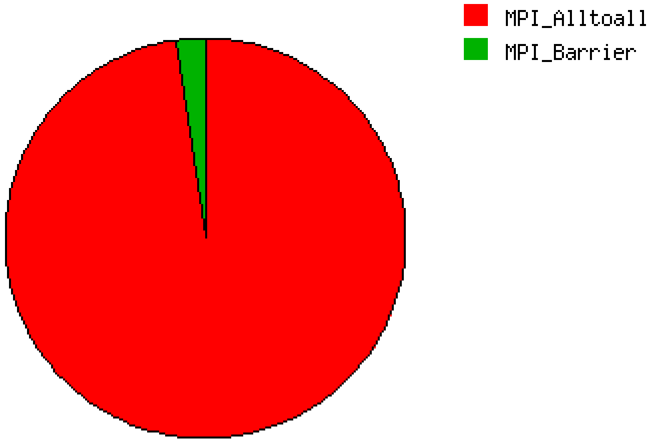


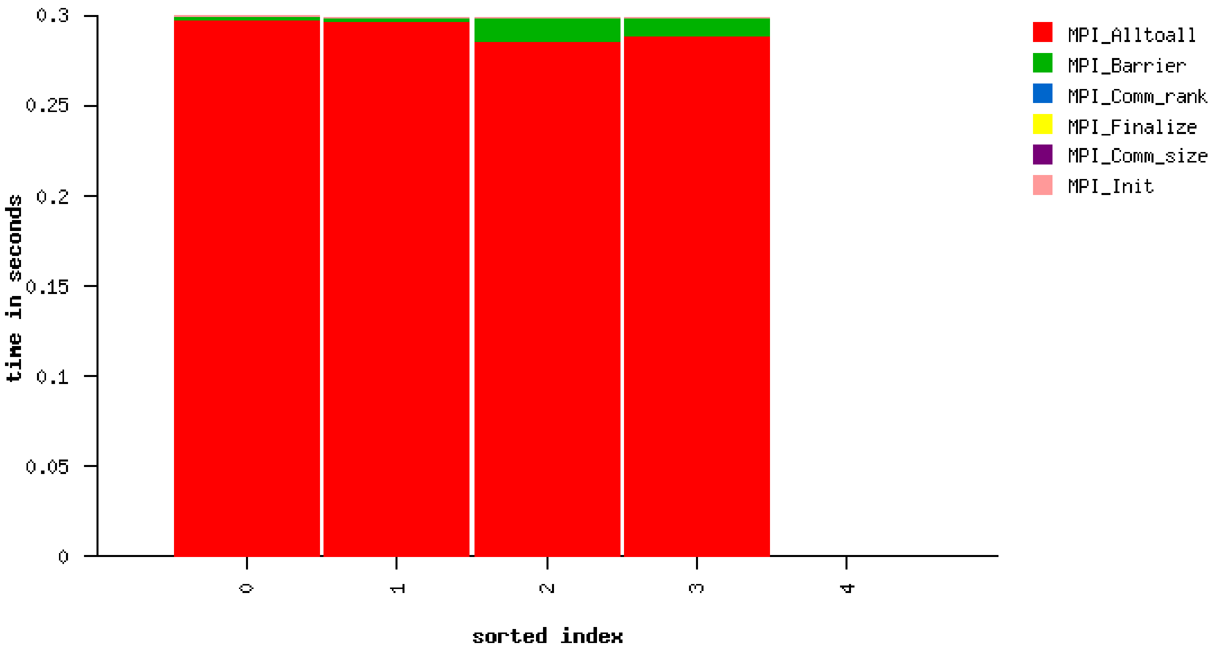


Max. barrier time: 0.2 ms

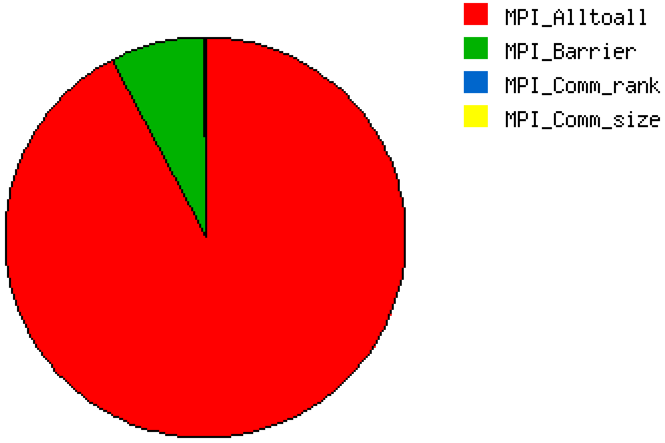
Alltoall - NPROCS=4, Data size = 4 MB

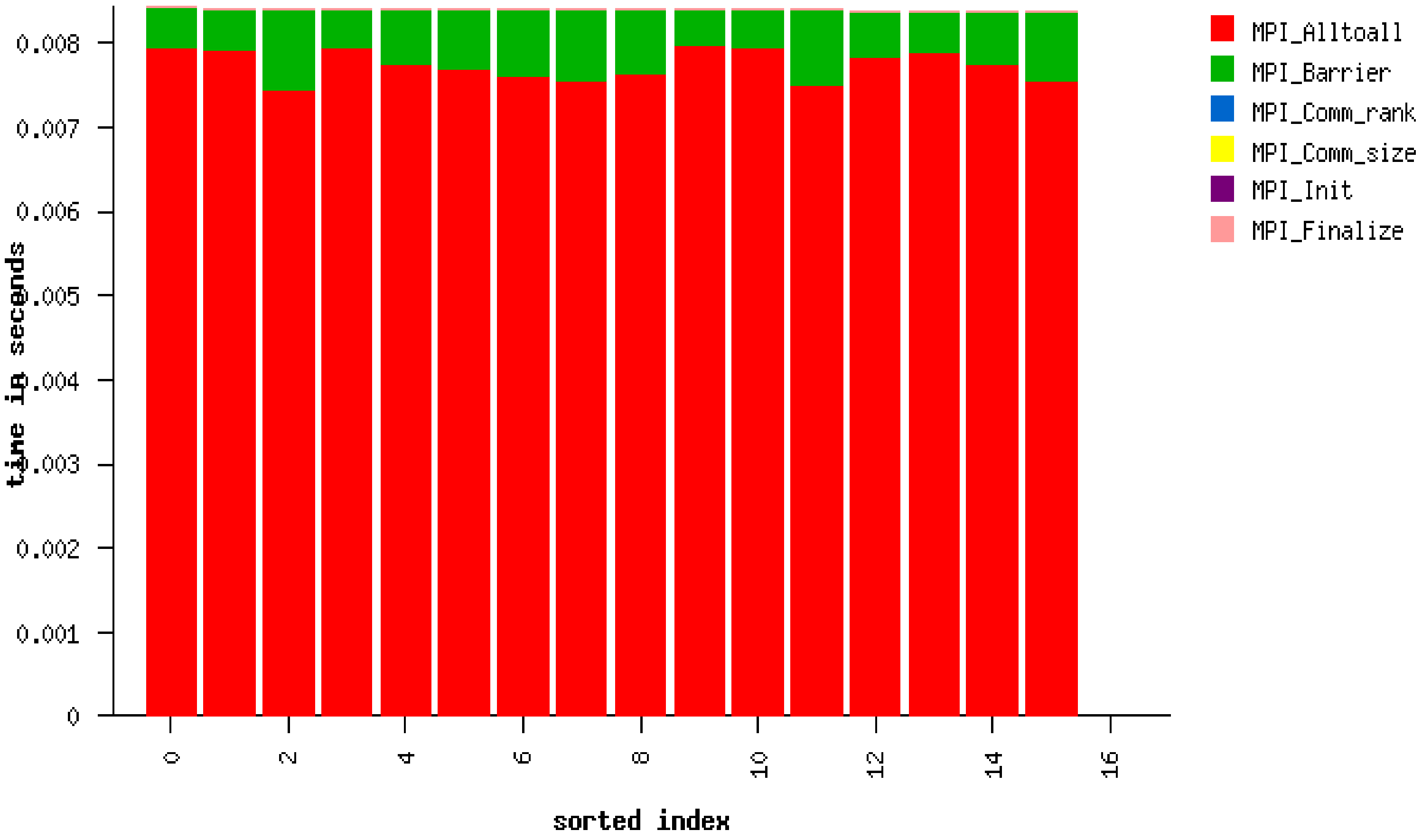
Max. barrier time: 1.2 ms
Max. Alltoall time: 7.8 ms



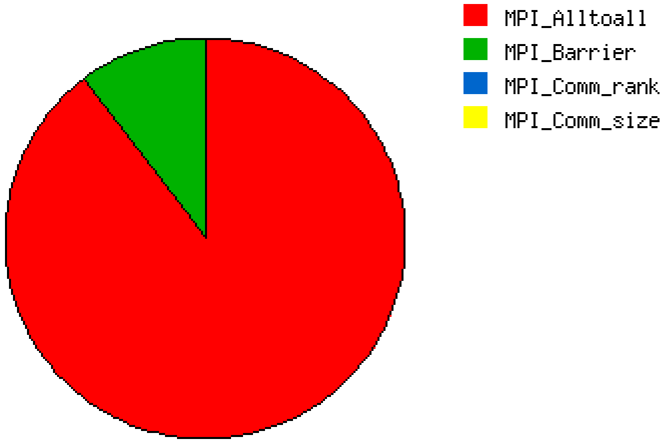


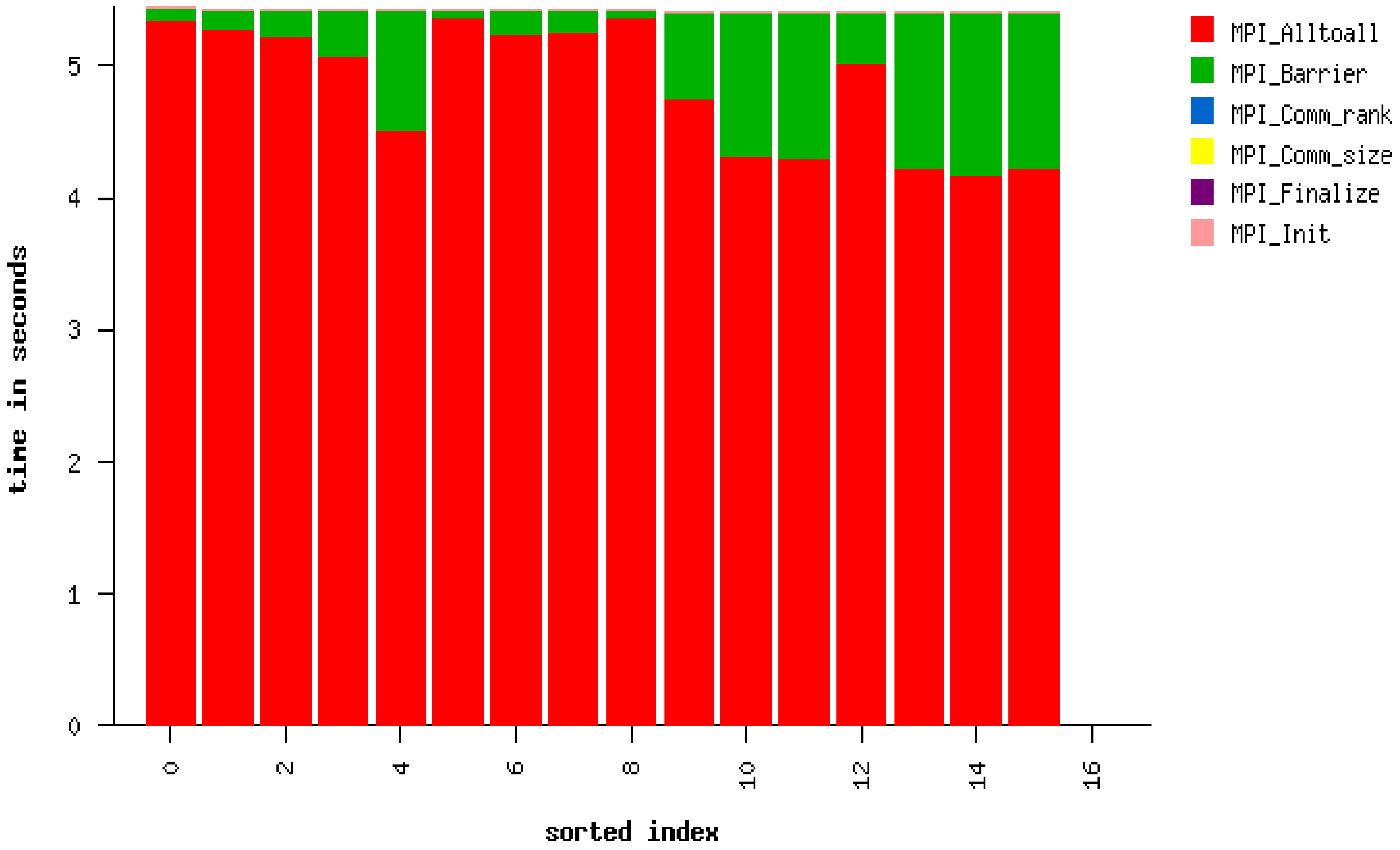
Alltoall - NPROCS=16, Data size = 4 KB



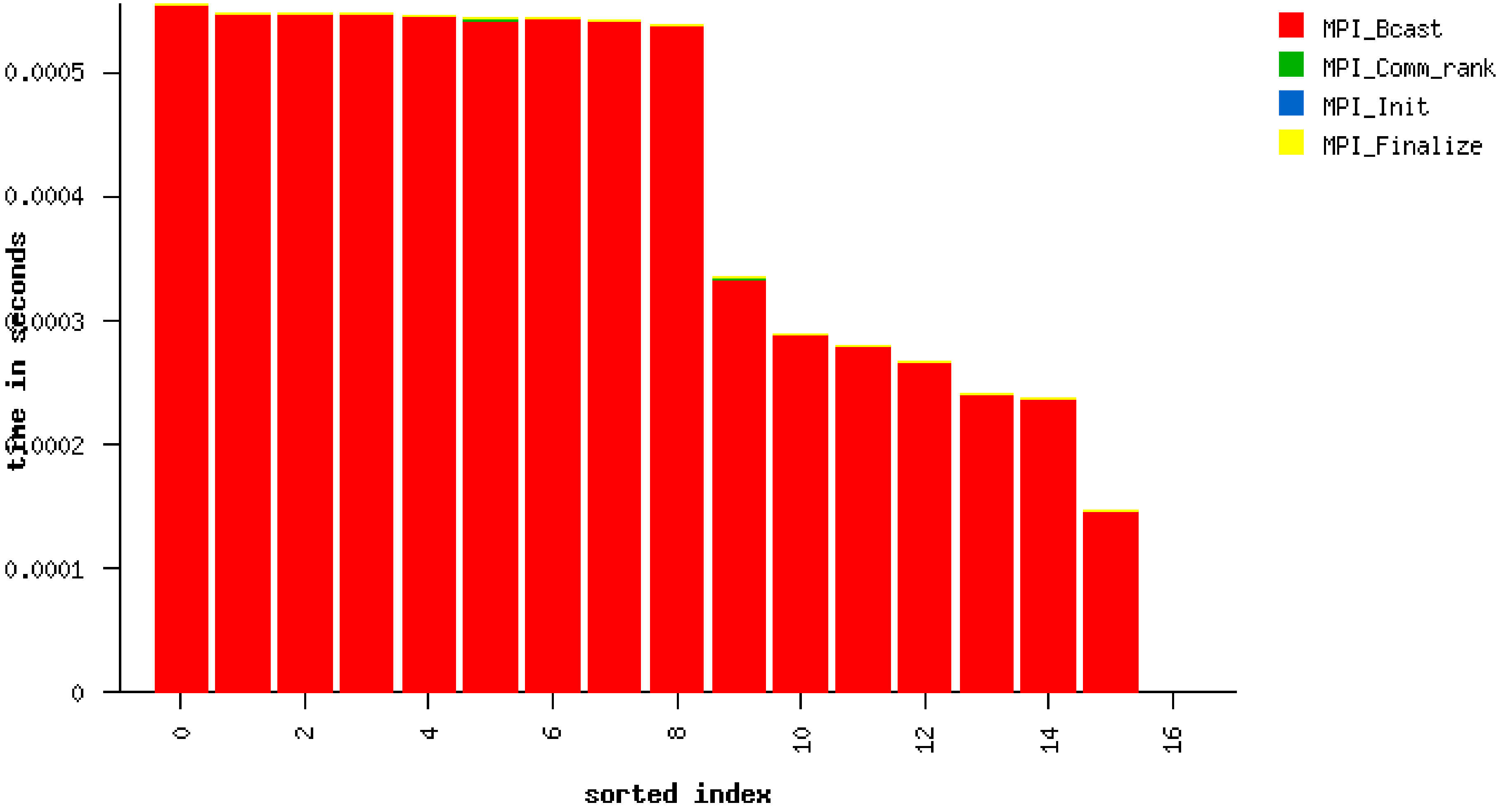


Alltoall - NPROCS=16, Data size = 4 MB

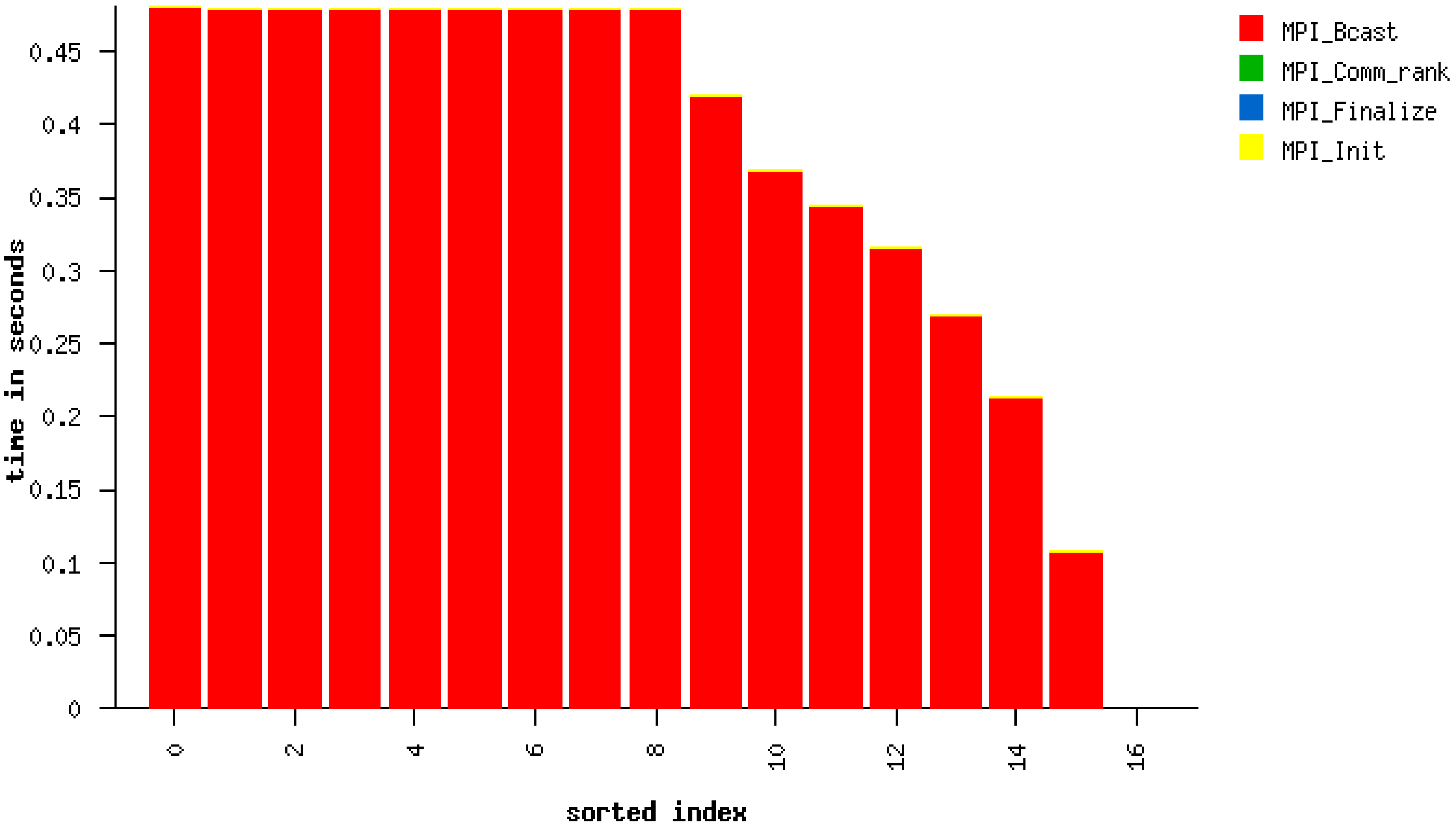




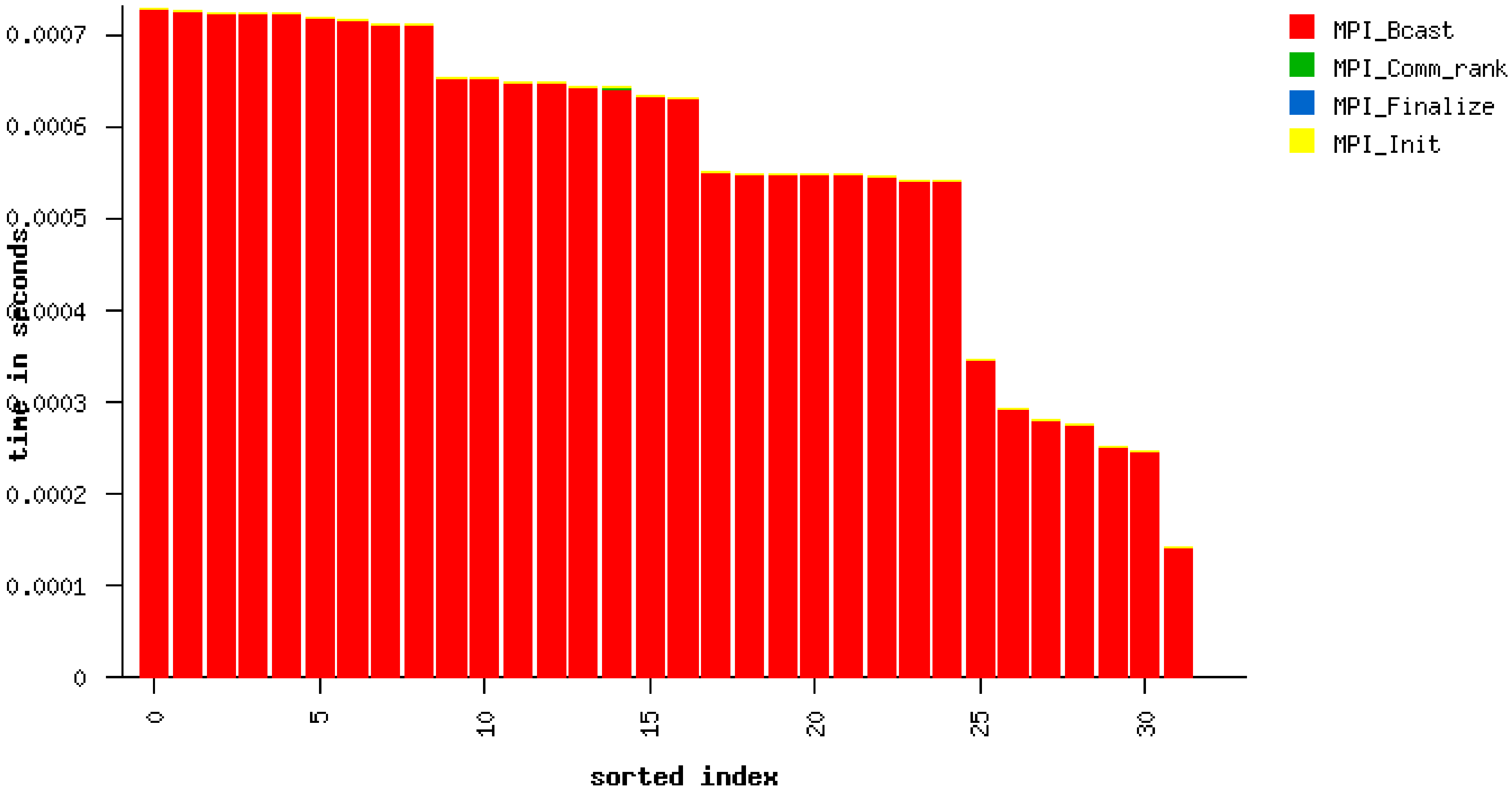
Bcast – 16P, 4 KB



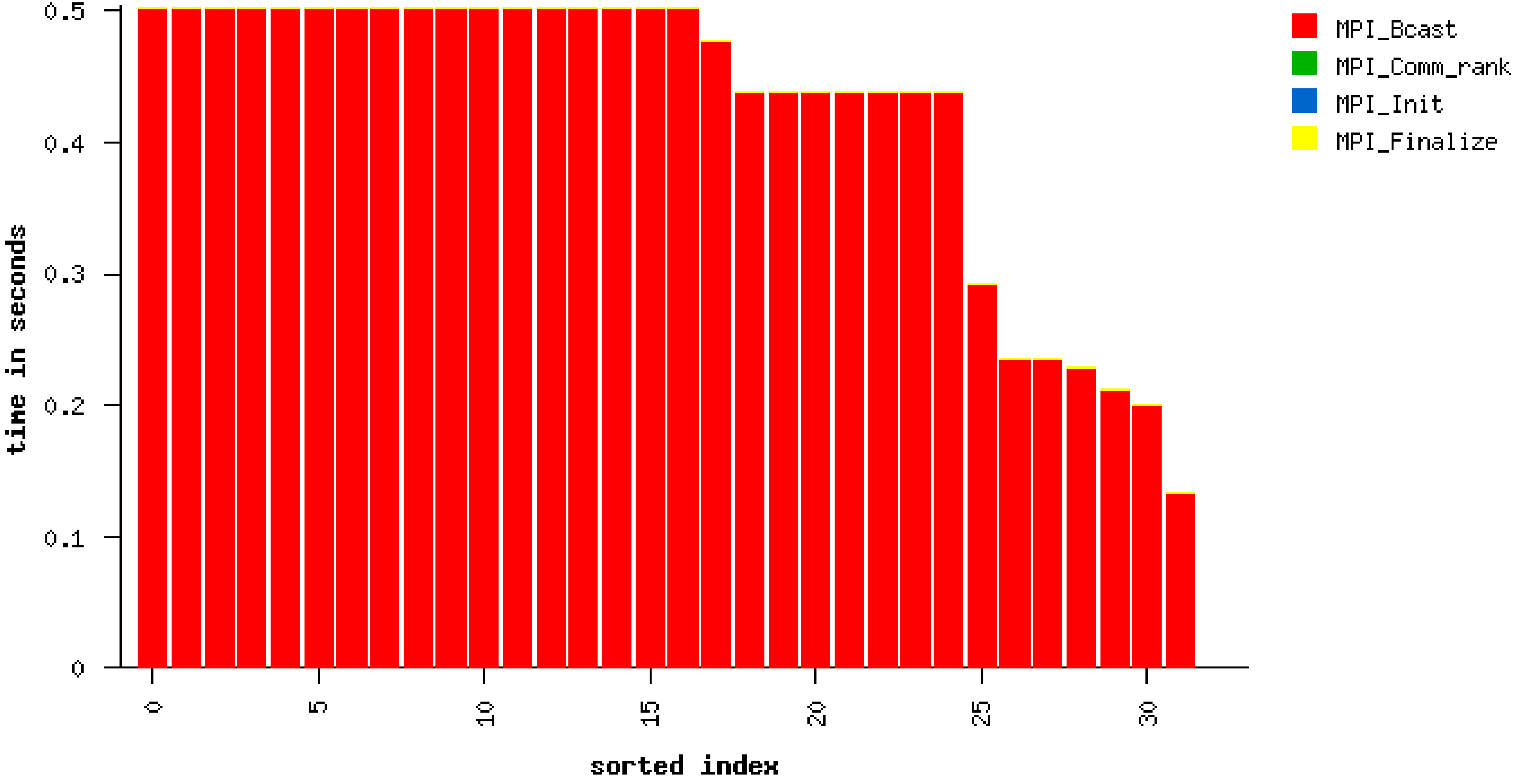
Bcast – 16P, 4 MB



Bcast – 32P, 4 KB

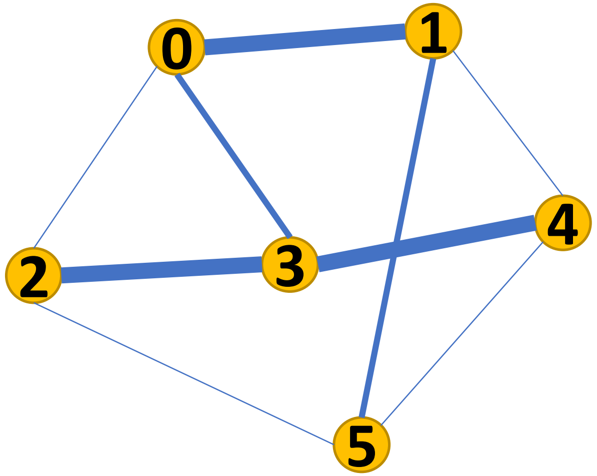


Bcast – 32P, 4 MB



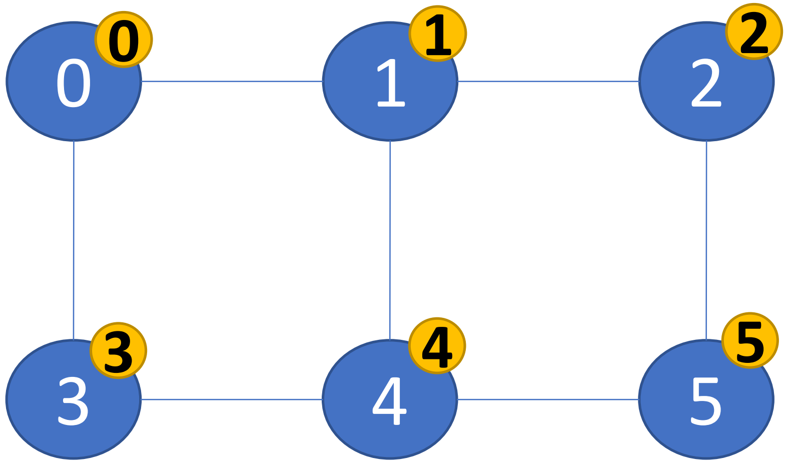
Revision

Communication Graph Mapping



	512		256		
512				64	256
			512		64
256		512		512	
	64		512		64
	256	64		64	

Linear mapping



Q1: What are the communicating pairs?

Q2: Distance/hops between the communicating pairs?

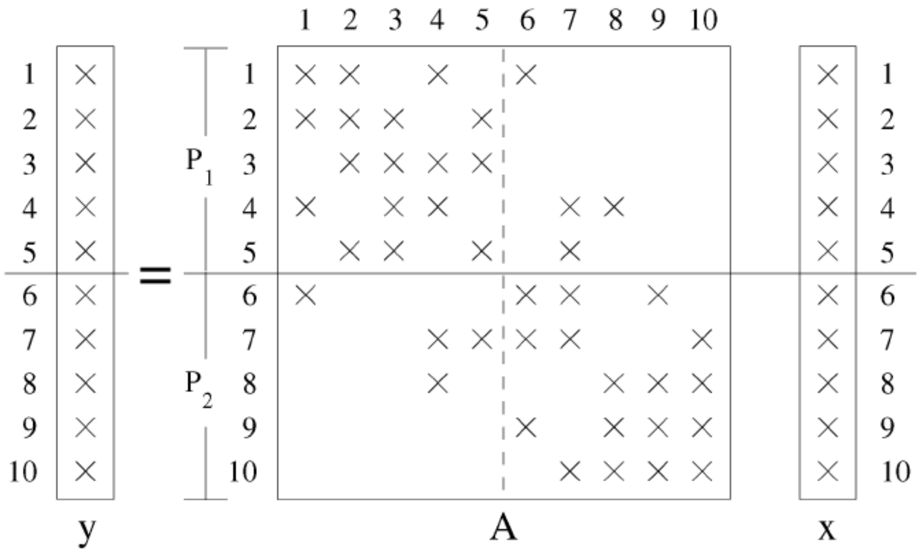
Q3: Total hop-bytes?

Estimation Function

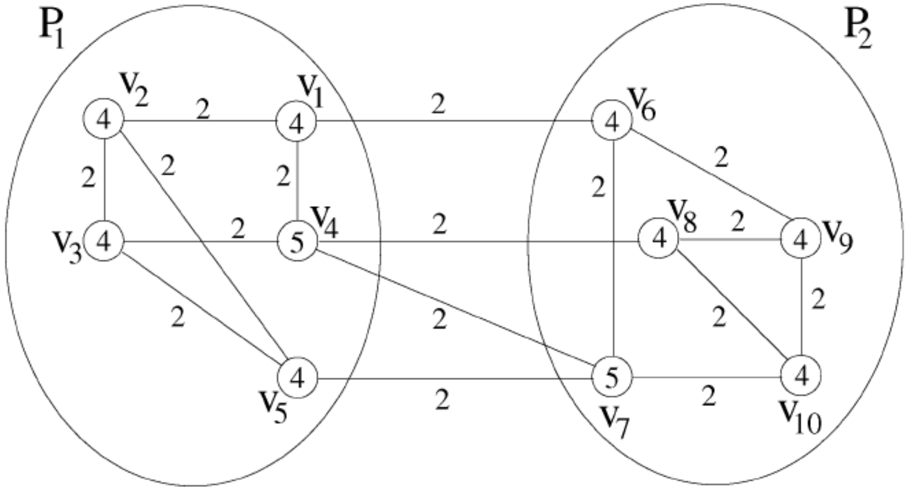
- $f_{est}(t, p, M) =$ Cost of placing a task t onto processor p under current task mapping M
- Estimate how critical it is to place a task in the current cycle, select the task with maximum criticality
- T_k is the set of tasks yet to be placed
- P_k is the set of processors that are available

$$\boxed{\begin{array}{l} T_k \cup \overline{T}_k = \emptyset \\ P_k \cup \overline{P}_k = \emptyset \end{array}}$$

Graph Model for SpMV



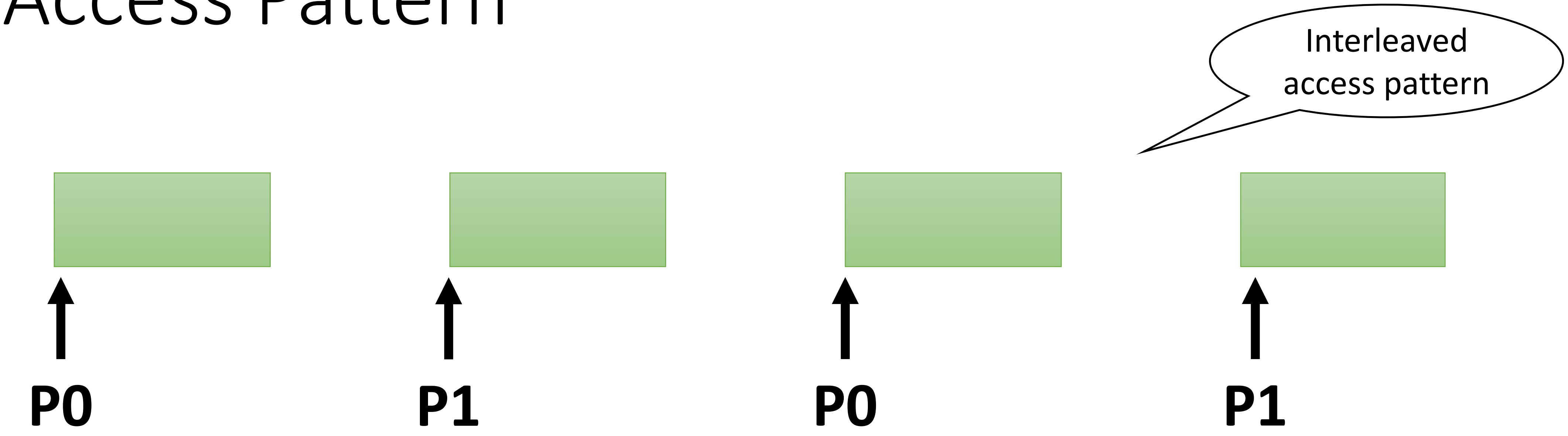
	1	2	3	4	5	6	7	8	9	10
1	×	×		×		×				
2	×	×	×		×					
3		×	×	×	×					
4	×		×	×			×	×		
5		×	×		×		×			
6	×					×	×		×	
7				×	×	×	×			×
8				×				×	×	×
9						×		×	×	×
10							×	×	×	×



- Computation load?
  - Similar for both processors
- Number of communications?
  - 8 as per this graph
  - Actually 6

Parallel I/O

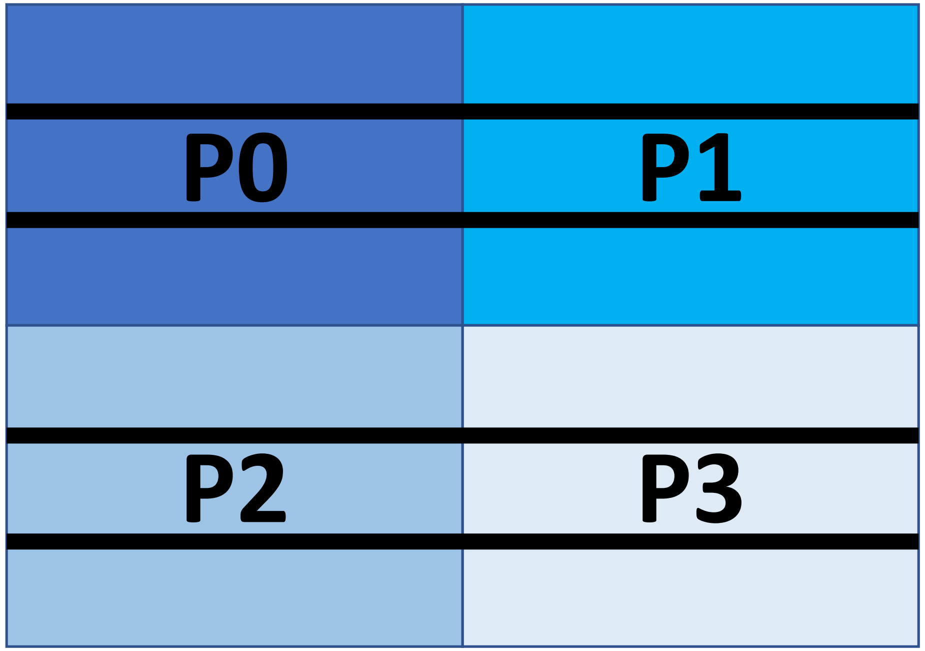
Access Pattern



Each process reads a small chunk of data from a common file

```
MPI_File_set_view (fh, displacement, etype, filetype, “native”, info)
MPI_File_read_all (fh, data, datacount, MPI_INT, status)
```

Multiple Non-contiguous Accesses



- Every process' local array is non-contiguous in file
- Every process needs to make small I/O requests
- Can these requests be merged?

Revision Q3: 3D domain decomposition

```
//initialize
for (int i=0; i<N; i++)
 for (int j=0; j<N; j++)
  for (int k=0; k<N; k++)
   data[i][j][k] = (rank+1) * (i+j+k);

int xStart=_______________________________,
yStart=_______________________________,
zStart=_______________________________;

int xEnd=_________________________________,
yEnd=________________________________,
zEnd=________________________________;
```

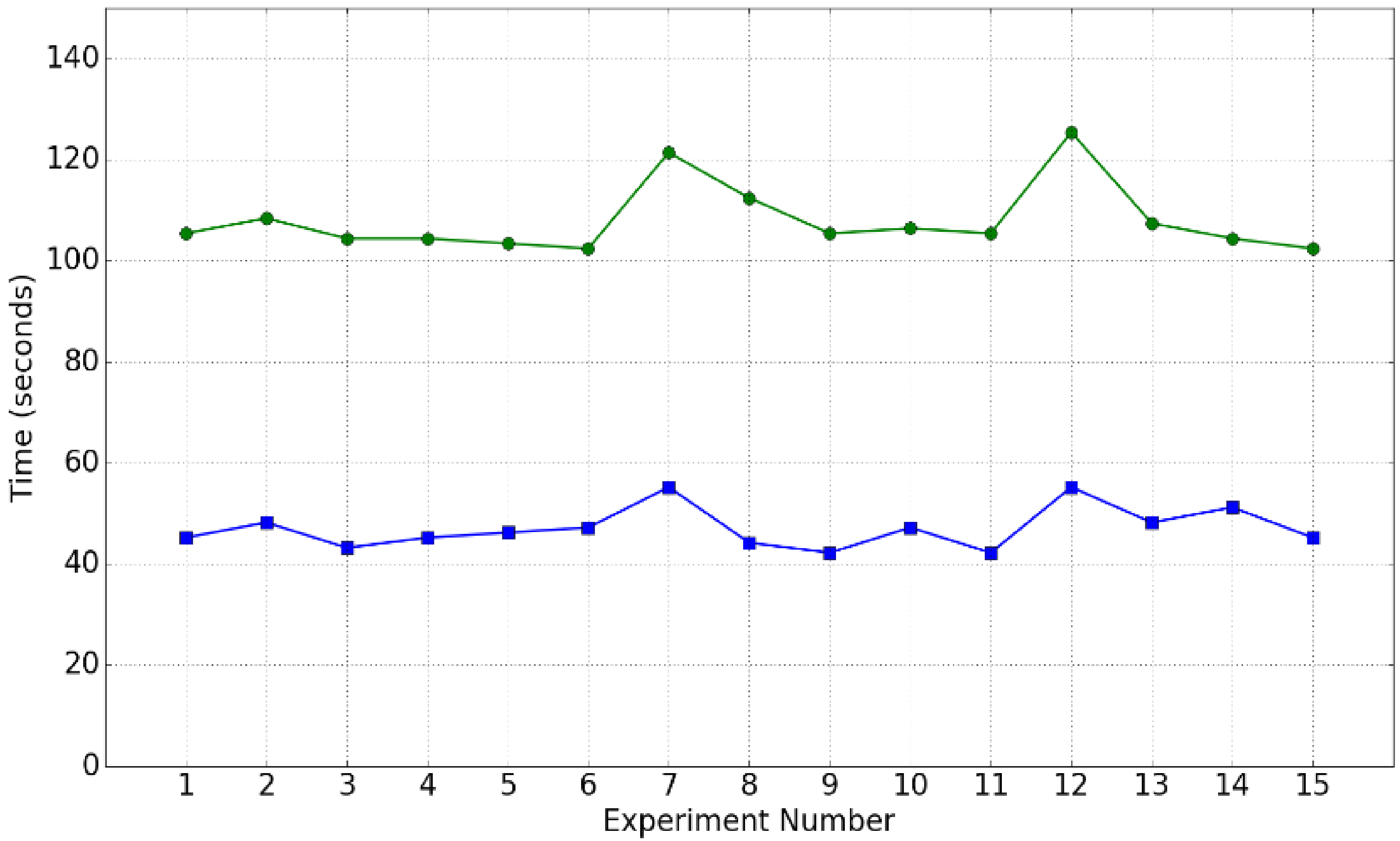
Revision Q4

A 3D matrix of size NxNxN was written to the file in the usual XYZ memory order. P processes read this 3D matrix from a file using parallel I/O following a 1D domain decomposition along Y-axis. Write an MPI code snippet for this (you may ignore the obvious initializations and finalizations). Assume that N is divisible by P.

Revision Q5

A sequential program P consists of three parts A, B, C. Part B is not parallelizable. Parts A and C are parallelizable. The sequential runtimes are Sa, Sb, Sc for the parts A, B, C respectively. Derive the speedup of P on N processes, where the overhead to parallelize part A is Oa, overhead to parallelize part C is Oc.

Revision Q6


Time (seconds)
0
20
40
60
80
100
120
140
Experiment Number
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15