HPC\_Chap3\_PartA

Chen Fang

**Problem 3.2**

local\_a = a + rank \* comm\_sz \* h

if( rank != comm\_sz-1 )

{

local\_n = (int)(n/comm\_sz)

local\_b = local\_a + local\_n \* h

}

else

{

local\_n = n % comm\_sz

local\_b = local\_a + local\_n \* h

}

**Problem 3.4**

if( my\_rank != 0 )

{

sprintf( message, “Proc %d of %d > Does anyone have a toothpick”, my\_rank, comm\_sz );

MPI\_Send( message, strlen(message)+1, MPI\_CHAR, 0, 0, MPI\_COMM\_WORLD );

}

else

{

printf( “Proc %d of %d > Does anyone have a toothpick”, my\_rank, comm\_sz );

for( int q = 1; q < comm\_sz; ++q )

{

MPI\_Recv( message, 100, MPI\_CHAR, q, 0, MPI\_COMM\_WORLD, MPI\_STATUS\_IGNORE );

printf( “%s\n”, message );

}

**Problem 3.5**

With enumeration, we have the following table:

|  |  |
| --- | --- |
| **Number of Leaves** | **Depth** |
| 1 | 0 |
| 2 | 1 |
| 4 | 2 |
| … | … |
| n/2 | log2(n/2)=log2(n)-1 |

As it is assumed that T is a complete tree, when depth is increased by one, the number of leaves will increase by 2\*number\_of\_nodes because each node has two leaves. So when number of nodes is n/2, the number of leaves is n while its depth is log2(n).

**Problem 3.6**

a)

|  |  |
| --- | --- |
| Block Distribution | |
| Processor | Components |
| 1 | 1 2 3 4 |
| 2 | 5 6 7 8 |
| 3 | 9 10 11 12 |
| 4 | 13 14 |

b)

|  |  |
| --- | --- |
| Cyclic Distribution | |
| Processor | Components |
| 1 | 1 5 9 13 |
| 2 | 2 6 10 14 |
| 3 | 3 7 11 |
| 4 | 4 8 12 |

c)

|  |  |
| --- | --- |
| Block-Cyclic Distribution (Block Size = 2) | |
| Processor | Components |
| 1 | 1 2 9 10 |
| 2 | 3 4 11 12 |
| 3 | 5 6 13 14 |
| 4 | 7 8 |