

# MPI\_Sendrecv

- MPI\_Send + MPI\_Recv

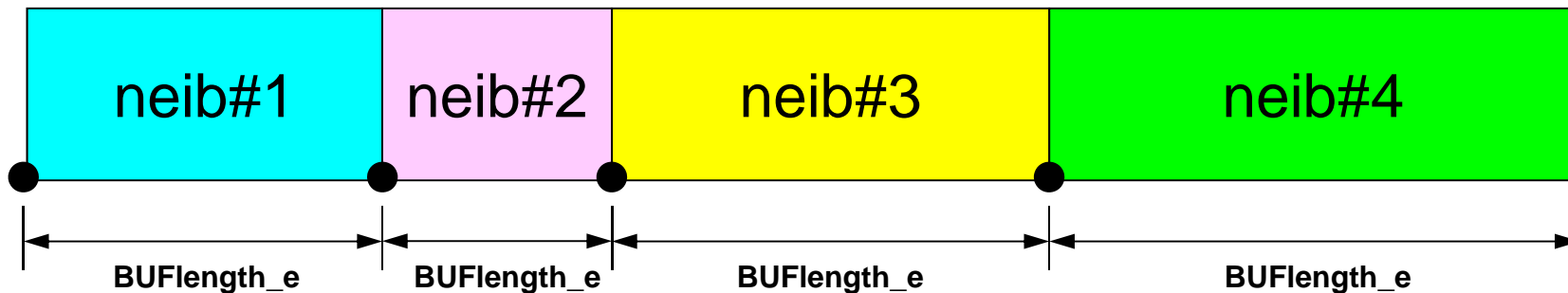
- call MPI\_Sendrecv

(**sendbuf, sendcount, sendtype, dest, sendtag, recvbuf, recvcount, recvtype, source, recvtag, comm, status, ierr**)

|   |                  |    |   |  |
|---|------------------|----|---|--|
| - | <u>sendbuf</u>   | 任意 | I | 送信バッファの先頭アドレス,   |
| - | <u>sendcount</u> | 整数 | I | 送信メッセージのサイズ  |
| - | <u>sendtype</u>  | 整数 | I | 送信メッセージのデータタイプ   |
| - | <u>dest</u>      | 整数 | I | 宛先プロセスのアドレス(ランク)   |
| - | <u>sendtag</u>   | 整数 | I | 送信用メッセージタグ, 送信メッセージの種類を区別するときを使用。<br>通常は「0」でよい。  |
| - | <u>recvbuf</u>   | 任意 | I | 受信バッファの先頭アドレス,   |
| - | <u>recvcount</u> | 整数 | I | 受信メッセージのサイズ  |
| - | <u>recvtype</u>  | 整数 | I | 受信メッセージのデータタイプ   |
| - | <u>source</u>    | 整数 | I | 送信元プロセスのアドレス(ランク)  |
| - | <u>recvtag</u>   | 整数 | I | 受信メッセージタグ, 送信メッセージの種類を区別するときを使用。<br>通常は「0」でよい。同じメッセージタグ番号同士で通信。                            |
| - | <u>comm</u>      | 整数 | I | コミュニケータを指定する   |
| - | <u>status</u>    | 整数 | O | 状況オブジェクト配列(配列サイズ: (MPI_STATUS_SIZE))<br>MPI_STATUS_SIZE: “mpif.h”で定められるパラメータ<br>C言語については後述 |
| - | <u>ierr</u>      | 整数 | O | 完了コード  |

# 送信 (MPI\_Sendrecv)

SENDbuf



export\_index(0)+1    export\_index(1)+1    export\_index(2)+1    export\_index(3)+1    export\_index(4)

```

do neib= 1, NEIBPETOT
  do k= export_index(neib-1)+1, export_index(neib)
    kk= export_item(k)
    SENDbuf(k)= VAL(kk)
  enddo
enddo

do neib= 1, NEIBPETOT
  iS_e= export_index(neib-1) + 1
  iE_e= export_index(neib )
  BUFlength_e= iE_e + 1 - iS_e

  call MPI_SENDRECV
&      (SENDbuf(iS_e), BUFlength_e, MPI_INTEGER, NEIBPE(neib), 0,&
&      RECVbuf(iS_i), BUFlength_i, MPI_INTEGER, NEIBPE(neib), 0,&
&      MPI_COMM_WORLD, stat_sr, ierr)
enddo

```

送信バッファへの代入

# 受信 (MPI\_Sendrecv)

```

do neib= 1, NEIBPETOT
  iS_i= import_index(neib-1) + 1
  iE_i= import_index(neib )
  BUFlength_i= iE_i + 1 - iS_i

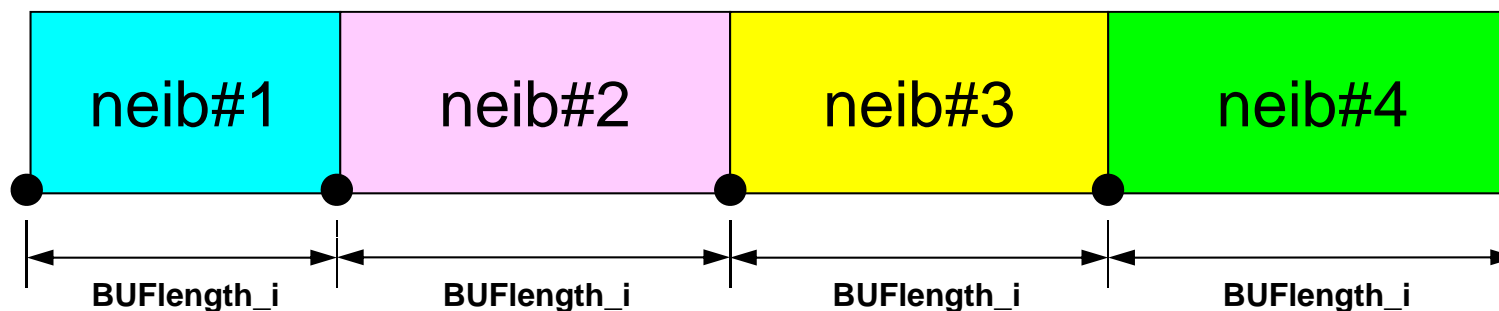
  call MPI_SENDRECV
&      (SENDbuf(iS_e), BUFlength_e, MPI_INTEGER, NEIBPE(neib), 0,&
&      RECVbuf(iS_i), BUFlength_i, MPI_INTEGER, NEIBPE(neib), 0,&
&      MPI_COMM_WORLD, stat_sr, ierr)
  enddo

do neib= 1, NEIBPETOT
  do k= import_index(neib-1)+1, import_index(neib)
    kk= import_item(k)
    VAL(kk)= RECVbuf(k)
  enddo
enddo

```

受信バッファからの代入

**RECVbuf**



import\_index(0)+1    import\_index(1)+1    import\_index(2)+1    import\_index(3)+1    import\_index(4)

# 3領域に分割

#PE2

|                       |                       |                       |                        |
|-----------------------|-----------------------|-----------------------|------------------------|
| <b>7</b><br><u>21</u> | <b>8</b><br><u>22</u> | <b>9</b><br><u>23</u> | <b>15</b><br><u>24</u> |
| <b>4</b><br><u>16</u> | <b>5</b><br><u>17</u> | <b>6</b><br><u>18</u> | <b>14</b><br><u>19</u> |
| <b>1</b><br><u>11</u> | <b>2</b><br><u>12</u> | <b>3</b><br><u>13</u> | <b>13</b><br><u>14</u> |
| <b>10</b><br><u>6</u> | <b>11</b><br><u>7</u> | <b>12</b><br><u>8</u> |                        |

#PE1

|                        |                       |                       |
|------------------------|-----------------------|-----------------------|
| <b>14</b><br><u>23</u> | <b>7</b><br><u>24</u> | <b>8</b><br><u>25</u> |
| <b>13</b><br><u>18</u> | <b>5</b><br><u>19</u> | <b>6</b><br><u>20</u> |
| <b>12</b><br><u>13</u> | <b>3</b><br><u>14</u> | <b>4</b><br><u>15</u> |
| <b>11</b><br><u>8</u>  | <b>1</b><br><u>9</u>  | <b>2</b><br><u>10</u> |
|                        | <b>9</b><br><u>4</u>  | <b>10</b><br><u>5</u> |

#PE0

|                        |                        |                        |                      |                        |
|------------------------|------------------------|------------------------|----------------------|------------------------|
| <b>11</b><br><u>11</u> | <b>12</b><br><u>12</u> | <b>13</b><br><u>13</u> |                      |                        |
| <b>6</b><br><u>6</u>   | <b>7</b><br><u>7</u>   | <b>8</b><br><u>8</u>   | <b>9</b><br><u>9</u> | <b>10</b><br><u>10</u> |
| <b>1</b><br><u>1</u>   | <b>2</b><br><u>2</u>   | <b>3</b><br><u>3</u>   | <b>4</b><br><u>4</u> | <b>5</b><br><u>5</u>   |

# MPI\_Sendrecvでうまく行くケース

#PE2

|         |         |         |          |
|---------|---------|---------|----------|
| 7<br>21 | 8<br>22 | 9<br>23 | 15<br>24 |
| 4<br>16 | 5<br>17 | 6<br>18 | 14<br>19 |
| 1<br>11 | 2<br>12 | 3<br>13 | 13<br>14 |
| 10<br>6 | 11<br>7 | 12<br>8 |          |

#PE1

|          |         |         |
|----------|---------|---------|
| 14<br>23 | 7<br>24 | 8<br>25 |
| 13<br>18 | 5<br>19 | 6<br>20 |
| 12<br>13 | 3<br>14 | 4<br>15 |
| 11<br>8  | 1<br>9  | 2<br>10 |
|          | 9<br>4  | 10<br>5 |

#PE0

|          |          |          |        |          |
|----------|----------|----------|--------|----------|
| 11<br>11 | 12<br>12 | 13<br>13 |        |          |
| 6<br>6   | 7<br>7   | 8<br>8   | 9<br>9 | 10<br>10 |
| 1<br>1   | 2<br>2   | 3<br>3   | 4<br>4 | 5<br>5   |

#PE0

```
#NEIBPEtot
  2
#NEIBPE
  1  2
```

#PE1

```
#NEIBPEtot
  2
#NEIBPE
  0  2
```

#PE2

```
#NEIBPEtot
  2
#NEIBPE
  1  0
```

|        | #PE0 | #PE1 | #PE2 |
|--------|------|------|------|
| neib=1 | 1    | 0    | 1    |
| neib=2 | 2    | 2    | 0    |

# MPI\_Sendrecvでblockingするケース

#PE2

|         |         |         |          |
|---------|---------|---------|----------|
| 7<br>21 | 8<br>22 | 9<br>23 | 15<br>24 |
| 4<br>16 | 5<br>17 | 6<br>18 | 14<br>19 |
| 1<br>11 | 2<br>12 | 3<br>13 | 13<br>14 |
| 10<br>6 | 11<br>7 | 12<br>8 |          |

#PE1

|          |         |         |
|----------|---------|---------|
| 14<br>23 | 7<br>24 | 8<br>25 |
| 13<br>18 | 5<br>19 | 6<br>20 |
| 12<br>13 | 3<br>14 | 4<br>15 |
| 11<br>8  | 1<br>9  | 2<br>10 |
|          | 9<br>4  | 10<br>5 |

#PE0

|          |          |          |        |          |
|----------|----------|----------|--------|----------|
| 11<br>11 | 12<br>12 | 13<br>13 |        |          |
| 6<br>6   | 7<br>7   | 8<br>8   | 9<br>9 | 10<br>10 |
| 1<br>1   | 2<br>2   | 3<br>3   | 4<br>4 | 5<br>5   |

#PE0

```
#NEIBPEtot
  2
#NEIBPE
  1  2
```

#PE1

```
#NEIBPEtot
  2
#NEIBPE
  2  0
```

#PE2

```
#NEIBPEtot
  2
#NEIBPE
  0  1
```

|        | #PE0 | #PE1 | #PE2 |
|--------|------|------|------|
| neib=1 | 1    | 2    | 0    |
| neib=2 | 2    | 0    | 1    |

# MPI\_Sendrecvでうまく行くケース

```

do neib= 1, NEIBPETOT
  iS_e= export_index(neib-1) + 1
  iE_e= export_index(neib )
  BUFlength_e= iE_e + 1 - iS_e

  call MPI_SENDRFCV
&      (SENDbuf(iS_e), BUFlength_e, MPI_INTEGER, NEIBPE(neib), 0,&
&      RECVbuf(iS_i), BUFlength_i, MPI_INTEGER, NEIBPE(neib), 0,&
&      MPI_COMM_WORLD, stat_sr, ierr)
enddo

```

|        | #PE0 | #PE1 | #PE2 |
|--------|------|------|------|
| neib=1 | 1    | 0    | 1    |
| neib=2 | 2    | 2    | 0    |

## ※完了状態

(#PE0, #PE1, #PE2) が、通信をそれぞれ完了している隣接領域数を表す。この問題であれば  
(2, 2, 2) となれば通信終了

## neib=1

- #PE0 : #PE1と通信
- #PE1 : #PE0と通信
- #PE2 : #PE1と通信
  - まず#PE0と#PE1が通信を行う, neib=2へ進む
  - #PE2は待ち状態, neib=1のまま, **完了状態 : (1, 1, 0)**

## neib=2

- #PE0 : #PE2と通信
- #PE1 : #PE2と通信
- #PE2 : #PE0と通信
  - #PE1@neib=2が#PE2@neib=1との通信を完了させる  
**完了状態 : (1, 2, 1)**
  - #PE0@neib=2と#PE2@neib=1が通信を行う  
**完了状態 : (2, 2, 2)**

# MPI\_Sendrecvでblockingするケース

```

do neib= 1, NEIBPETOT
  iS_e= export_index(neib-1) + 1
  iE_e= export_index(neib )
  BUFlength_e= iE_e + 1 - iS_e

  call MPI_SENDRFCV
&      (SENDbuf(iS_e), BUFlength_e, MPI_INTEGER, NEIBPE(neib), 0,&
&      RECVbuf(iS_i), BUFlength_i, MPI_INTEGER, NEIBPE(neib), 0,&
&      MPI_COMM_WORLD, stat_sr, ierr)
enddo

```

|        | #PE0 | #PE1 | #PE2 |
|--------|------|------|------|
| neib=1 | 1    | 2    | 0    |
| neib=2 | 2    | 0    | 1    |

## neib=1

- #PE0 : #PE1と通信
- #PE1 : #PE2と通信
- #PE2 : #PE0と通信
  - マッチする通信相手が居ない
  - neib=1を完了できず次へ進めない : blocking

## ※完了状態

(#PE0, #PE1, #PE2) が、通信をそれぞれ完了している隣接領域数を表す。この問題であれば  
(2, 2, 2) となれば通信終了