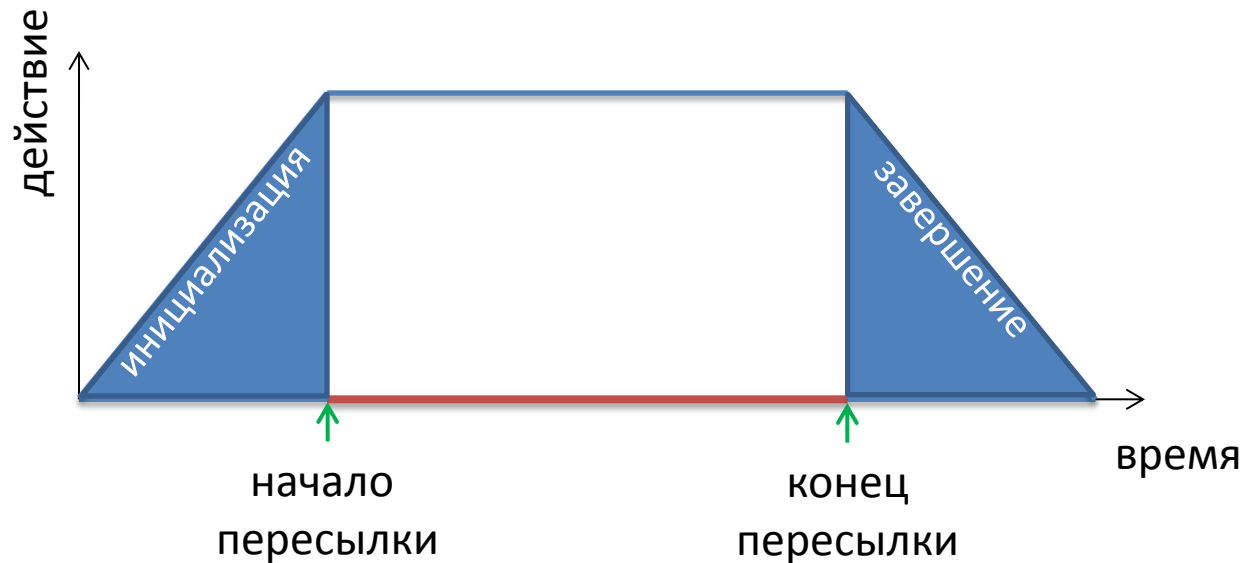
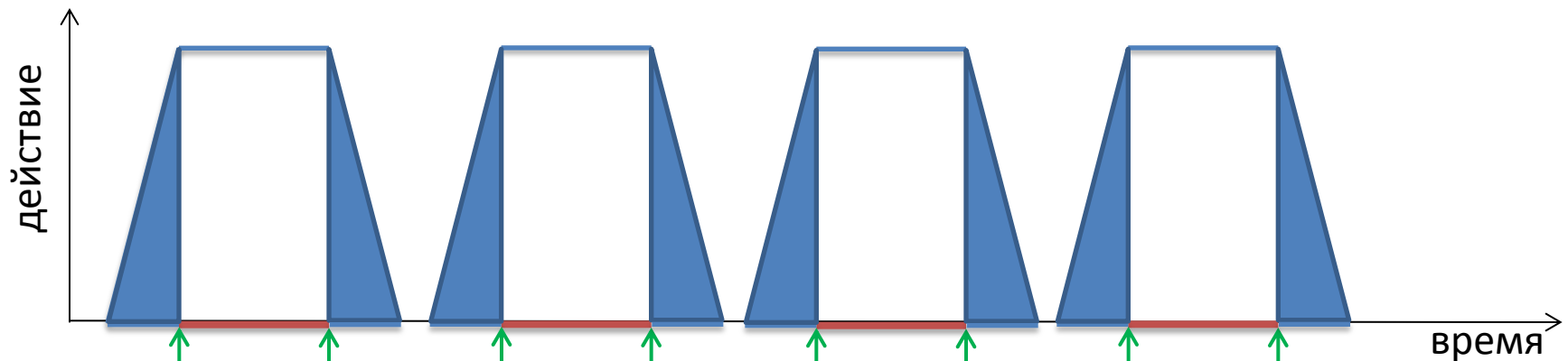


Передавать одно большое или много маленьких?

Временная диаграмма пересылки одного сообщения:

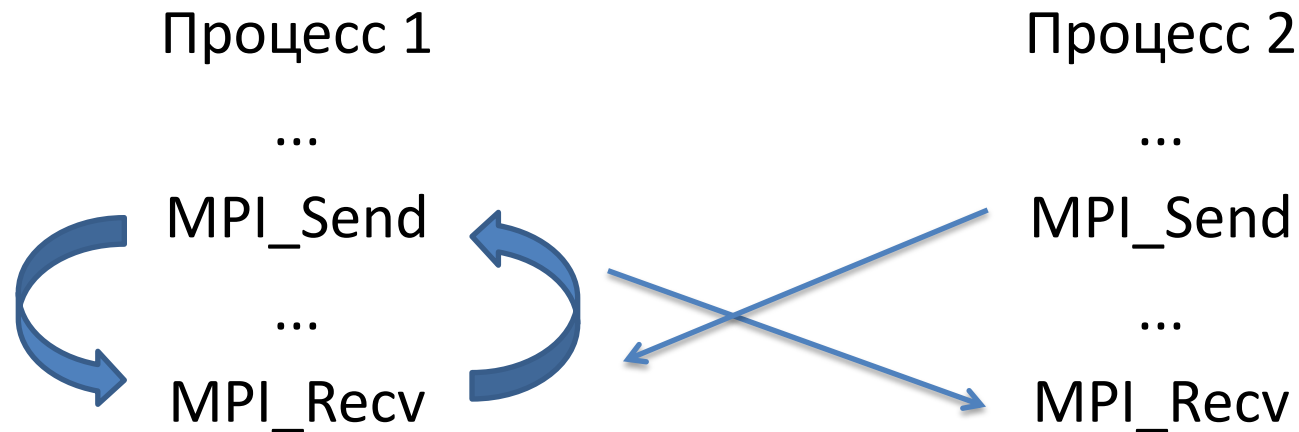


Временная диаграмма пересылки многих сообщений:



Классический тупик

Возникает при использовании блокирующих функций передачи и приема сообщений



Как избежать:

- 1) Смена мест операторов
- 2) Использование неблокирующих функций MPI_Isend и MPI_Irecv + оператор MPI_Wait

Передача сообщений без блокировки

Возврат из функций происходит сразу после начала передачи. Переменная Request идентифицирует пересылку.

```
MPI_Request Request;
```

```
MPI_Status Status;
```

```
...
```

```
MPI_Isend(&array[5], 5, MPI_INT, 1, 1, MPI_COMM_WORLD,  
&Request) ;
```

```
...
```

```
MPI_Irecv(&array[0], 5, MPI_INT, 0, 1, MPI_COMM_WORLD,  
&Request) ;
```

```
...
```

```
MPI_Wait(&Request, &Status); /*Блокирующая операция.
```

```
Возврат происходит после завершения операции, связанной  
с запросом request. В параметре status возвращается  
информация о законченной операции.*/
```

Классический тупик

Еще один способ как избежать тупик:

3) Использование функции MPI_Sendrecv

```
MPI_Sendrecv(&array[5], 5, MPI_INT, 1, 1,    // передача  
             &buf[0], 7, MPI_FLOAT, 0, 0,    // прием  
             MPI_COMM_WORLD, &Status) ;
```

Буферы передачи и приема не должны пересекаться!

- Гарантируется, что тупика не возникнет.
- Сообщение, отправленное операцией MPI_Sendrecv, можно принять обычным способом
- Сообщение, отправленное обычным способом, можно принять с помощью MPI_Sendrecv

Определение времени работы параллельной программы

`double MPI_Wtime(void)` – возвращает астрономическое время в секундах (вещественное число), произошедшее с некоторого момента в прошлом. Разность возвращаемых значений покажет время работы данного участка.

Пример:

```
double begin, end, total;  
begin = MPI_Wtime();  
....  
end = MPI_Wtime();  
total = end – begin;
```