

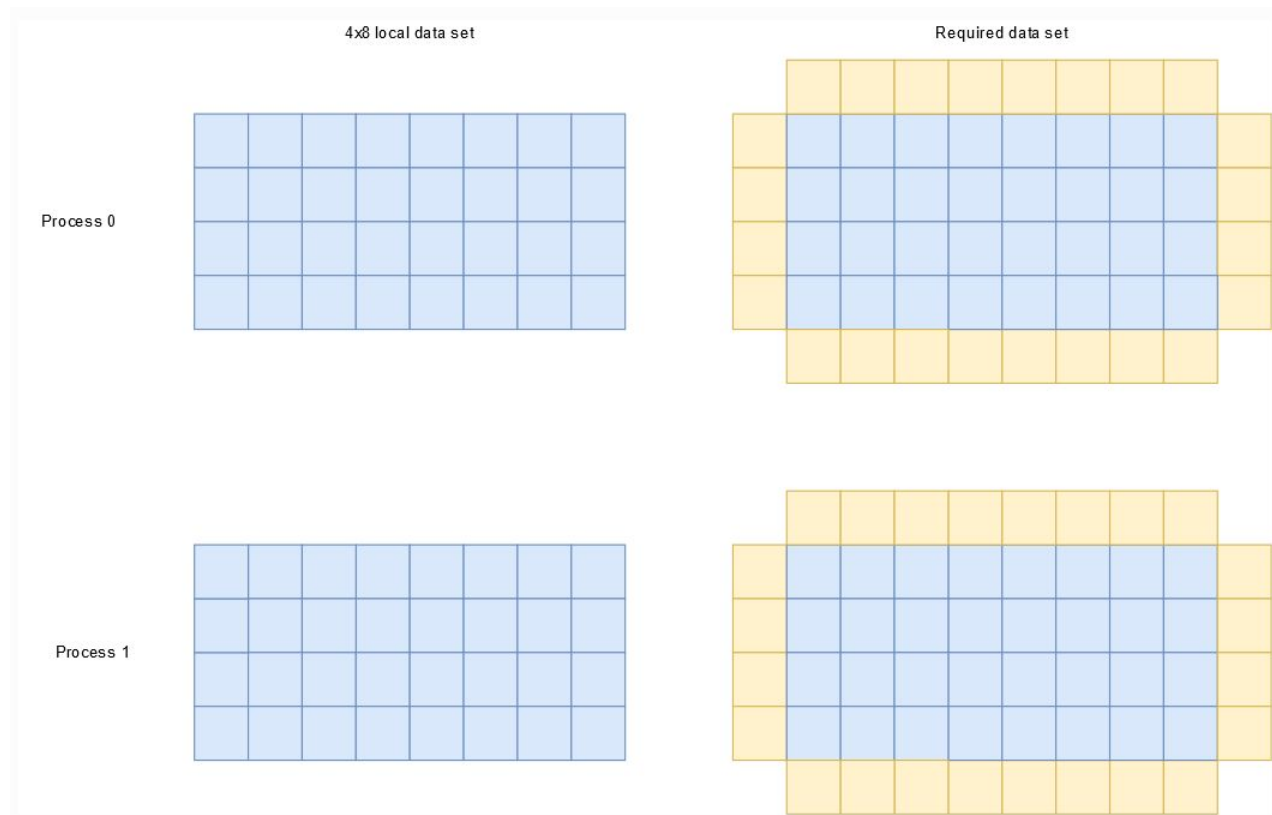
Средства и системы параллельного программирования

#8. MPI: collectives & non-blocking

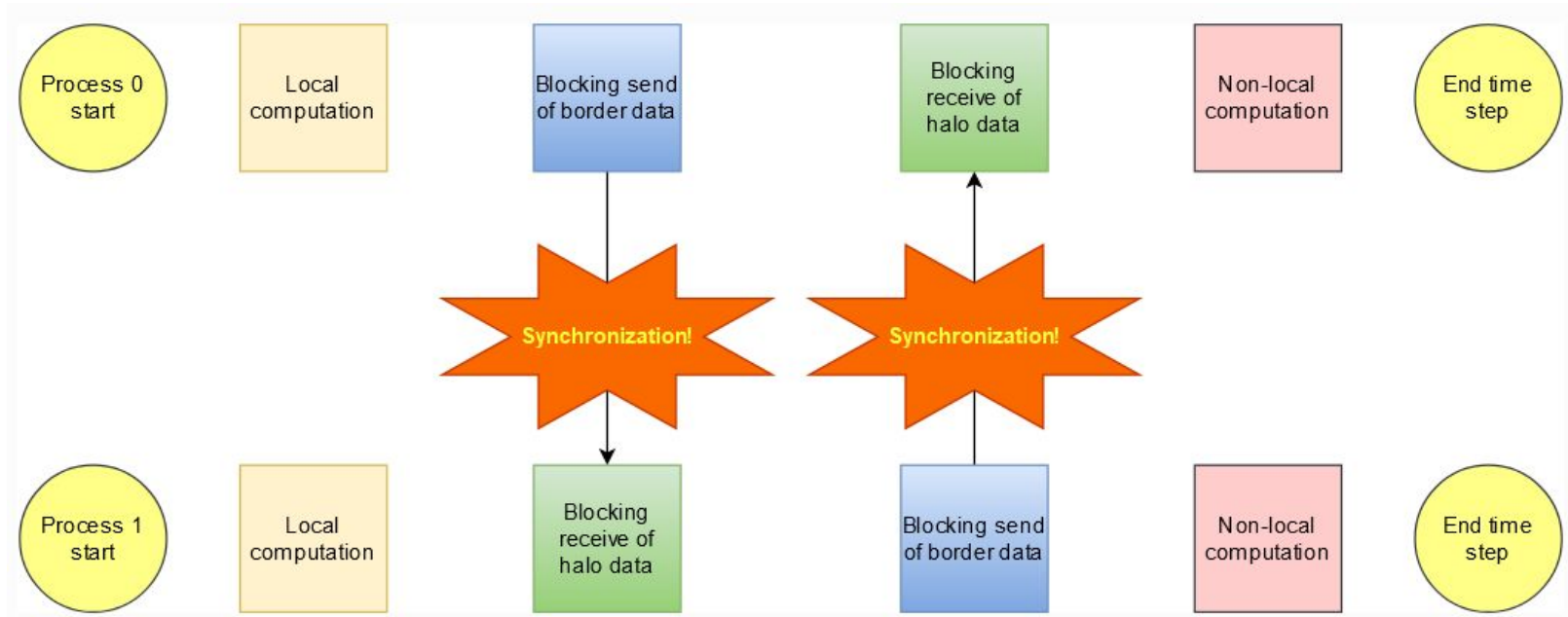
Режимы передачи сообщений в MPI

Communication Mode	Blocking Routines	Nonblocking Routines
Synchronous	MPI_Ssend	MPI_Issend
Buffered	MPI_Bsend	MPI_Ibsend
Ready	MPI_Rsend	MPI_Irsend
Standard	MPI_Send	MPI_Isend
	MPI_Recv	MPI_Irecv
	MPI_Sendrecv	
	MPI_Sendrecv_replace	

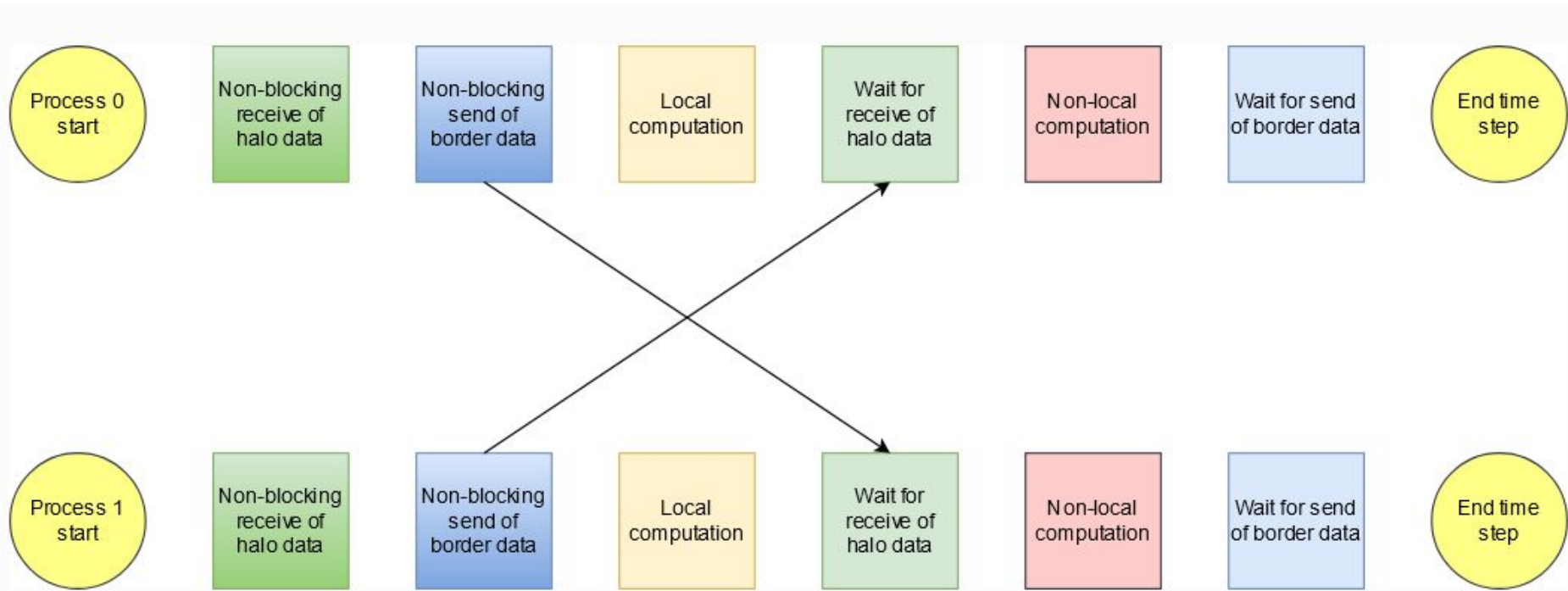
Сетка с halo-областью



Blocking

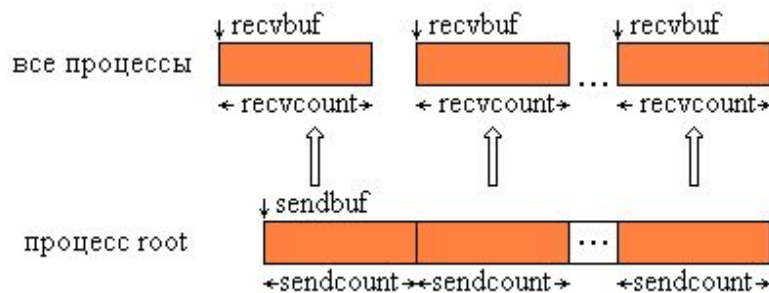


Non-blocking

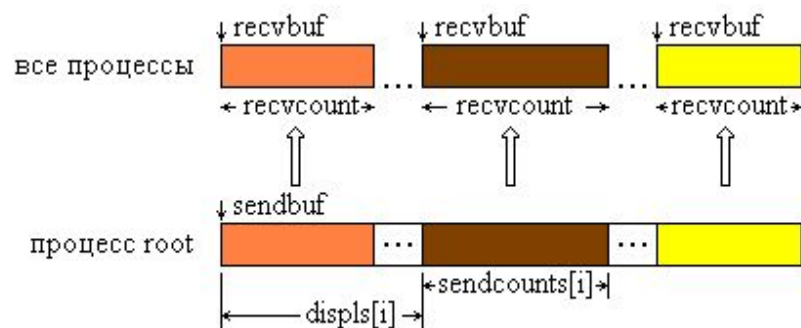


MPI Scatter

```
int MPI_Scatter(void* sendbuf, int  
sendcount, MPI_Datatype sendtype,  
void* recvbuf, int recvcount,  
MPI_Datatype recvtype, int root,  
MPI_Comm comm)
```



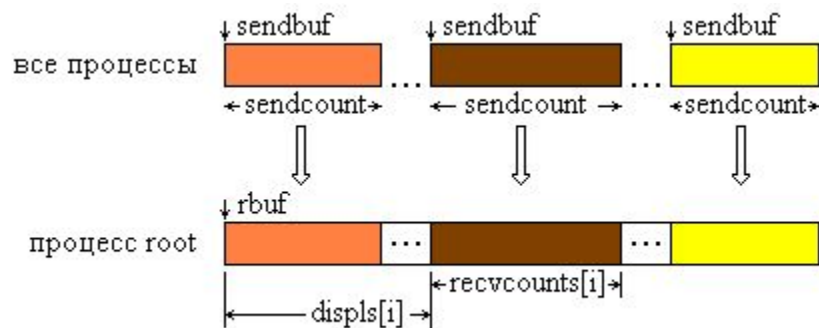
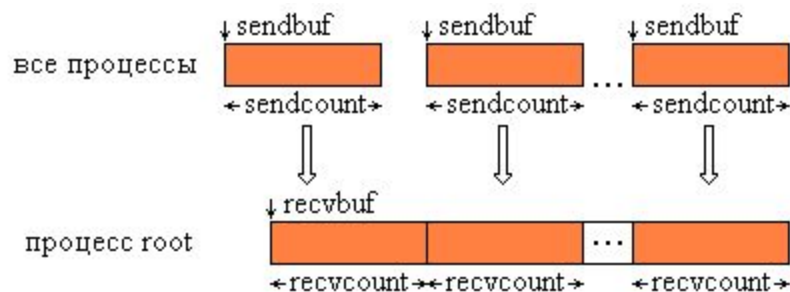
```
int MPI_Scatterv(void* sendbuf,  
int *sendcounts, int *displs,  
MPI_Datatype sendtype, void*  
recvbuf, int recvcount,  
MPI_Datatype recvtype, int root,  
MPI_Comm comm)
```



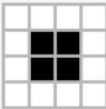
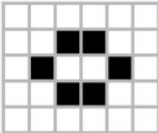
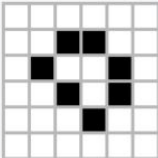
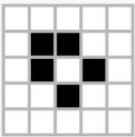
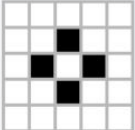
MPI Gather

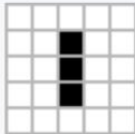
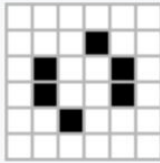
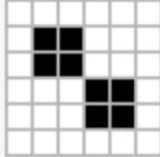
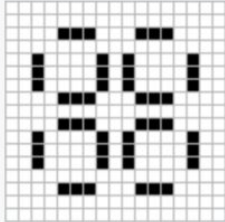
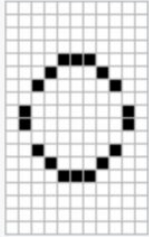
```
int MPI_Gather (void* sendbuf,  
int sendcount, MPI_Datatype  
sendtype,  
void* recvbuf, int recvcount,  
MPI_Datatype recvtype,  
int root, MPI_Comm comm)
```

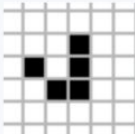
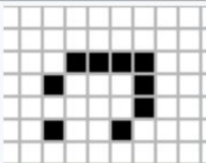
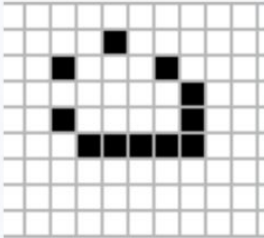
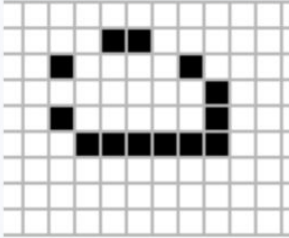
```
int MPI_Gatherv(void* sendbuf,  
int sendcount, MPI_Datatype  
sendtype,  
void* rbuf, int *recvcounts, int  
*displs, MPI_Datatype recvtype,  
int root, MPI_Comm comm)
```



Задание

Still lifes	
Block	
Bee-hive	
Loaf	
Boat	
Tub	

Oscillators	
Blinker (period 2)	
Toad (period 2)	
Beacon (period 2)	
Pulsar (period 3)	
Penta-decathlon (period 15)	

Spaceships	
Glider	
Light-weight spaceship (LWSS)	
Middle-weight spaceship (MWSS)	
Heavy-weight spaceship (HWSS)	

Задание

[Conway's Game of Life](#). Правила:

1. Живая клетка меньше чем с двумя живыми клетками-соседями умирает из-за малонаселённости
2. Живая клетка с двумя или тремя живыми клетками-соседями остаётся жить на следующей итерации
3. Живая клетка более чем с тремя живыми клетками-соседями умирает из-за перенаселённости
4. Неживая клетка с тремя клетками-соседями оживает

Задание

Написать программу с использованием MPI, реализующую игру в жизнь.

- 1) Инициализировать начальное расположение живых клеток **с помощью коллективных операций**, метод инициализации произвольный.
(например, можно сгенерировать начальные точки структур-глайдеров)
- 2) Провести K итераций игры
- 3) Начиная с K-ой итерации, если на каком-либо процессе число живых клеток на i -ой и $i+1$ -ой итерациях игры совпадает, остановить игру на всей клеточной области, вывести общее число живых клеток в конце.
Коммуникацию в данном случае также вести с помощью **коллективных операций**.

Задание

Требования (прим. сетка = весь набор клеток)

- 1) Программу запускать на Polus, через mpisubmit.pl
- 2) Для обмена данными между процессами использовать **неблокирующие** операции MPI.
- 3) Допускается (но не обязательно) иметь циклическую сетку (т.е. граничные точки сетки являются соседями друг для друга).
- 4) Хранить в памяти процесса только ту часть сетки, которая соответствует построенному разбиению
- 5) Разбиение сетки по процессам - произвольное (можно использовать ленточное)
- 6) Составить *краткий* отчёт, нарисовать $T(P)$, $P = \{1, 2, 4, 8, 16\}$.

Дедлайн: 25.11, 2.12