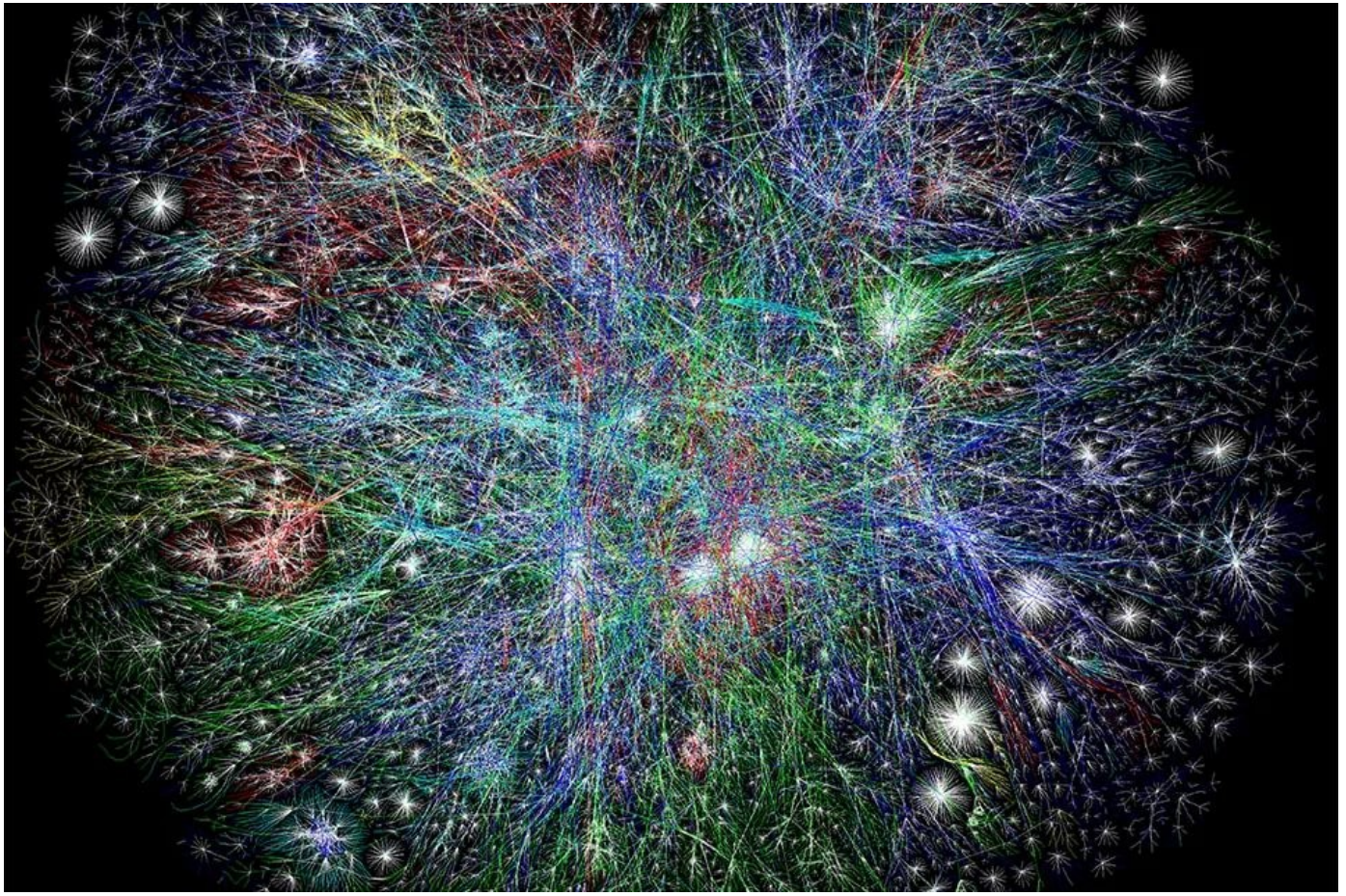
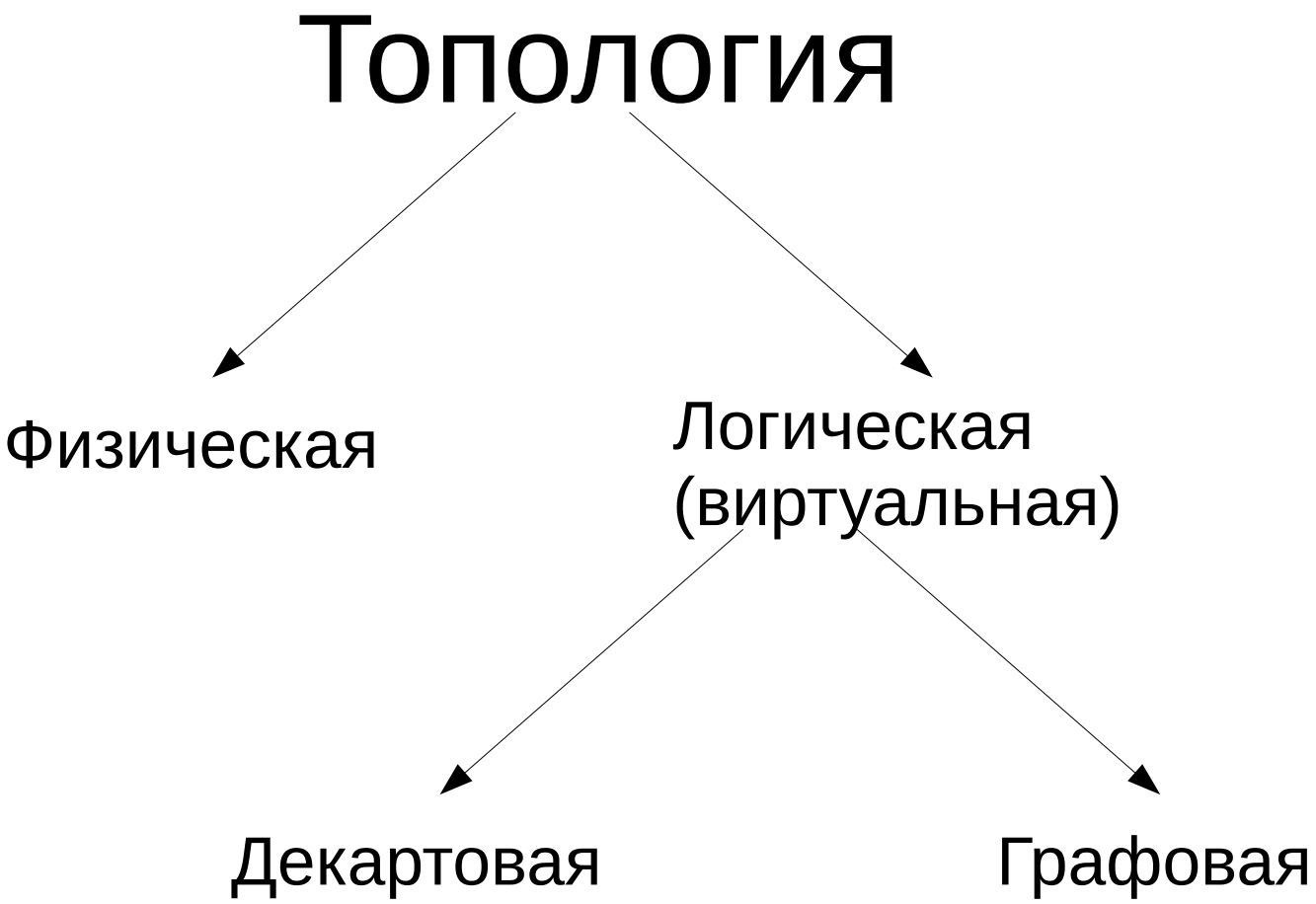
Виртуальные топологии процессов



Рак Алексей

Процессы параллельной программы объединяются в **группы**. В группу могут входить все процессы параллельной программы; с другой стороны, в группе может находиться только часть имеющихся процессов. Соответственно, один и тот же процесс может принадлежать нескольким группам. Управление группами процессов предпринимается для создания на их основе **коммуникаторов**.

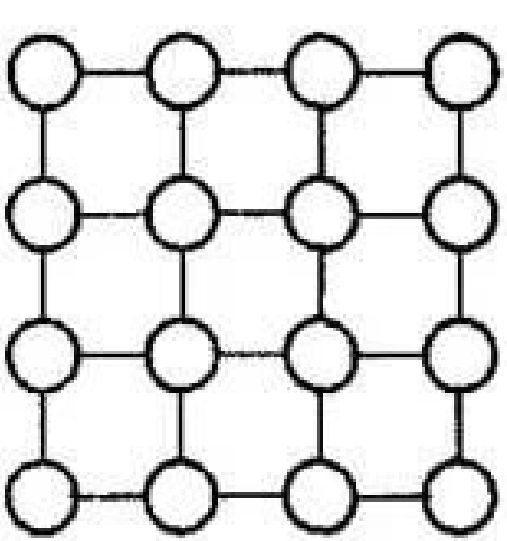
Под **коммуникатором** в MPI понимается специально создаваемый служебный объект, объединяющий в своем составе группу процессов и ряд дополнительных параметров (**контекст**), используемых при выполнении операций передачи данных. Как правило, парные операции передачи данных выполняются для процессов, принадлежащих одному и тому же коммуникатору. Коллективные операции применяются одновременно для всех процессов коммуникатора. Создание коммуникаторов предпринимается для уменьшения области действия коллективных операций и для устранения взаимовлияния разных выполняемых частей параллельной программы. Важно еще раз подчеркнуть – коммуникационные операции, выполняемые с использованием разных коммуникаторов, являются независимыми и не влияют друг на друга.


Топология
Физическая
Логическая
(виртуальная)
Декартовая
Графовая

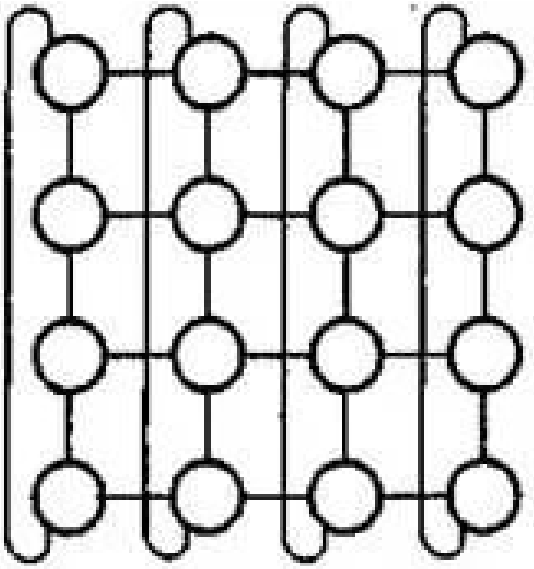
Виртуальная топология определяет специальное отображение номеров процессов в группе на определенную топологию, и наоборот.

Использование виртуальных процессов может оказаться полезным в силу ряда разных причин. Виртуальная топология, например, может больше соответствовать имеющейся структуре линий передачи данных. Применение виртуальных топологий может заметно упростить в ряде случаев представление и реализацию параллельных алгоритмов.

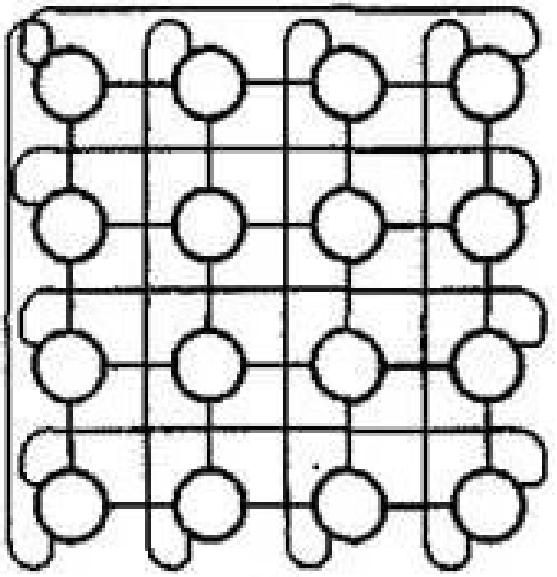
Декартова топология



1



2



3

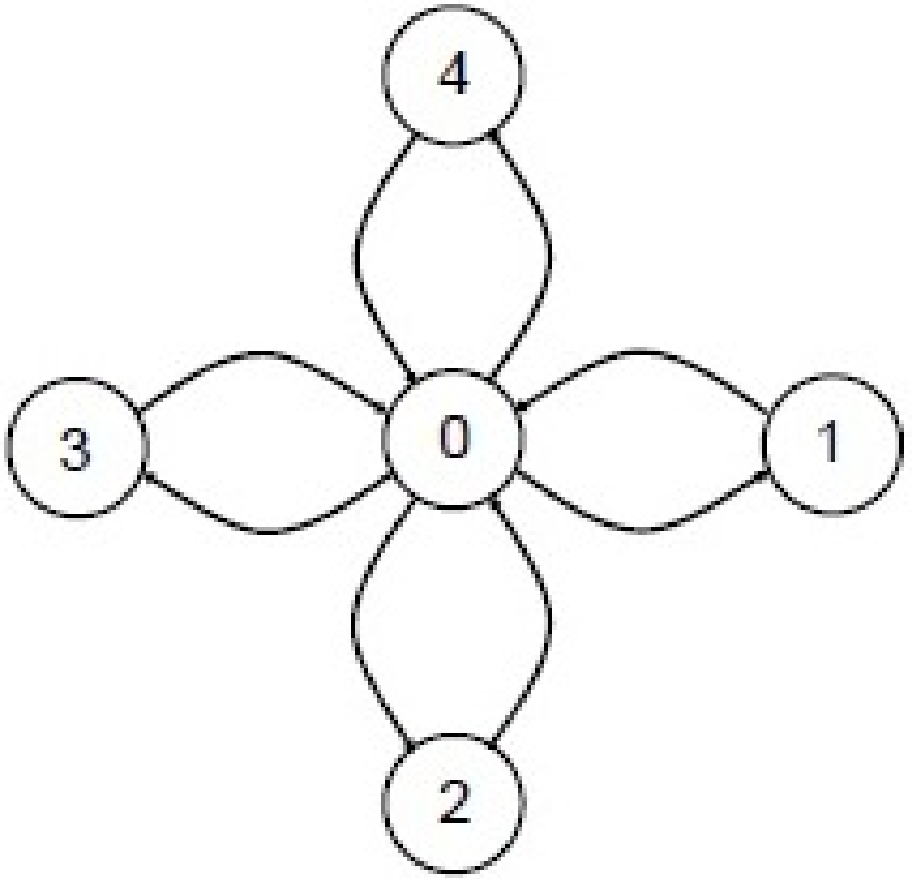
MPI_CART_CREATE(comm_old, ndims, dims, periods, reorder, comm_cart)

IN	comm_old	исходный коммуникатор (дескриптор)
IN	ndims	размерность создаваемой декартовой решетки (целое)
IN	dims	целочисленный массив размера ndims, хранящий количество процессов по каждой координате
IN	periods	массив логических элементов размера ndims, определяющий, периодична (true) или нет (false) решетка в каждой размерности
IN	reorder	нумерация может быть сохранена (false) или переупорядочена (true) (логическое значение)
OUT	comm_cart	коммуникатор новой декартовой топологии (дескриптор)

MPI_DIMS_CREATE(nnodes, ndims, dims)

IN	nnodes	количество узлов решетки (целое)
IN	ndims	число размерностей декартовой решетки(целое)
INOUT	dims	целочисленный массив размера ndims, указывающий количество вершин в каждой размерности.

Графовая топология



MPI_GRAPH_CREATE(comm_old, nnodes, index, edges, reorder, comm_graph)

IN	comm_old	входной коммуникатор (дескриптор)
IN	nnodes	количество узлов графа (целое)
IN	index	массив целочисленных значений, описывающий степени вершин
IN	edges	массив целочисленных значений, описывающий ребра графа
IN	reorder	номера могут быть переупорядочены (true) или нет (false)
OUT	comm_graph	построенный коммуникатор с графовой топологией (дескриптор)

MPI_TOPO_TEST(comm, status)

IN	comm	коммуникатор (дескриптор)
OUT	status	тип топологии коммуникатора comm (альтернатива)

Значения принимаемые status

MPI_GRAPH	топология графа
MPI_CART	декартова топология
MPI_UNDEFINED	топология не определена

Топологические функции запроса

```
int MPI_Graphdims_get(MPI_Comm comm, int *nnodes, int *nedges)

int MPI_Graph_get(MPI_Comm comm, int maxindex, int maxedges,
                  int *index, int *edges)

int MPI_Cartdim_get(MPI_Comm comm, int *ndims)

int MPI_Cart_get(MPI_Comm comm, int maxdims, int *dims,
                 int *periods, int *coords)

int MPI_Cart_rank(MPI_Comm comm, int *coords, int *rank)

int MPI_Cart_coords(MPI_Comm comm, int rank, int maxdims,
                    int *coords)

int MPI_Graph_neighbors_count(MPI_Comm comm, int rank,
                              int *nneighbors)

int MPI_Graph_neighbors(MPI_Comm comm, int rank,
                        int maxneighbors, int *neighbors)
```

Сдвиг в декартовых координатах

MPI_CART_SHIFT(comm, direction, disp, rank_source, rank_dest)

IN	comm	коммуникатор с декартовой топологией (дескриптор)
IN	direction	координата сдвига (целое)
IN	disp	направление смещения (> 0: смещение вверх, < 0: смещение вниз) (целое)
OUT	rank_source	номер процесса-отправителя (целое)
OUT	rank_dest	номер процесса-получателя (целое)

Декомпозиция декартовых структур

MPI_CART_SUB(comm, remain_dims, newcomm)

IN	comm	коммуникатор с декартовой топологией (дескриптор)
IN	remain_dims	i-й элемент в remain_dims показывает, содержится ли i-я размерность в подрешетке (true) или нет (false) (вектор логических элементов)
OUT	newcomm	коммуникатор, содержащий подрешетку, которая включает вызываемый процесс (дескриптор)

Разделение декартовых топологий

```
int MPI_Cart_sub(MPI_Comm comm_cart, int *remain_dims,
                 MPI_Comm *comm_slice);
```