Межведомственный суперкомпьютерный центр Российской академии наук –
филиал Федерального государственного учреждения «Федеральный научный центр
Научно-исследовательский институт системных исследований Российской академии наук»
(МСЦ РАН – филиал ФГУ ФНЦ НИИСИ РАН)

Рыбаков Алексей Анатольевич

Внутреннее представление и механизм межпроцессного обмена для блочно-структурированной сетки при выполнении расчетов на суперкомпьютере

Национальный Суперкомпьютерный Форум (НСКФ) 2016
29.11-02.12.2016, ИПС имени А. К. Айламазяна, Переславль-Залесский, Россия

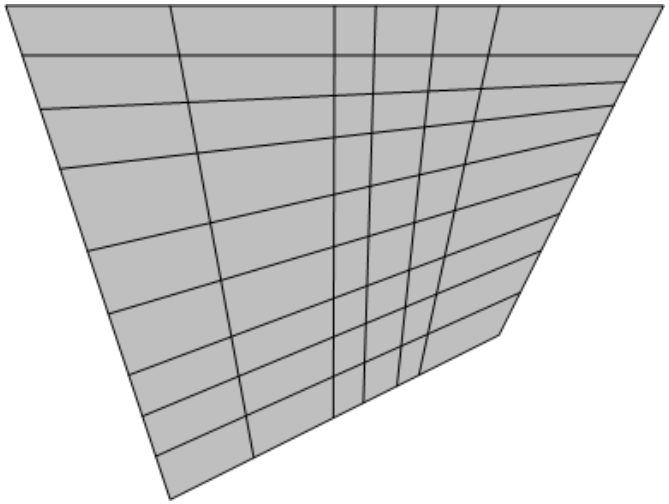
Использование суперкомпьютеров для численного моделирования

Суперкомпьютерный кластер МВС-10П на базе микропроцессоров Intel Xeon и сопроцессоров Intel Xeon Phi, и МВС-10П МП на основе массивно-параллельной архитектуры RSC PetaStream.

Численные расчеты задач газовой динамики как правило связаны с большим объемом вычислений, поэтому для таких задач актуально использование суперкомпьютеров.

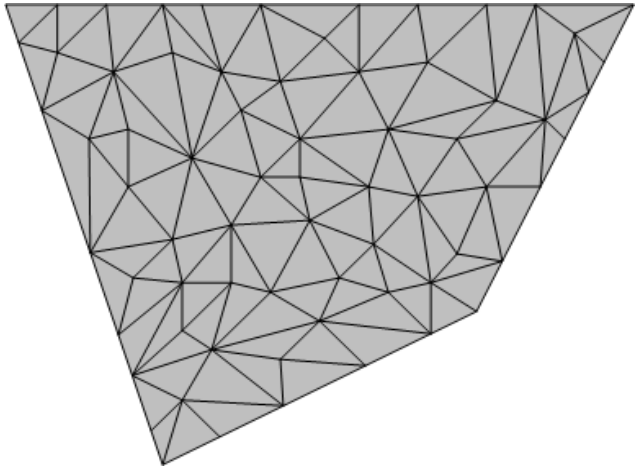
Использование блочно-структурированных сеток в расчетах задач газовой динамики

структурированная сетка



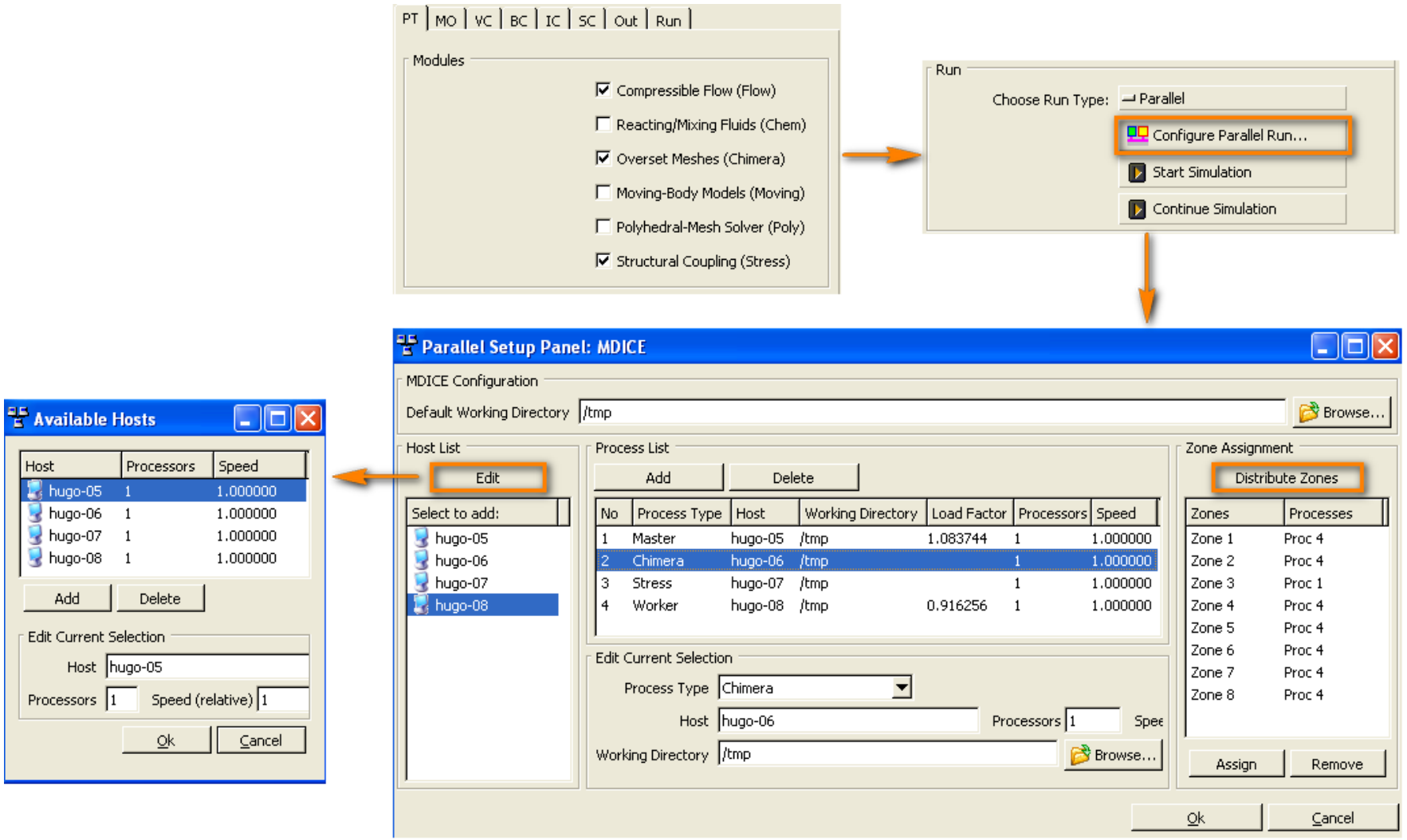
VS

неструктурированная сетка



В расчетах задач газовой динамики часто используются блочно-структурированные сетки, состоящие из структурированных блоков. Ячейки упорядочены внутри структурированного блока, что позволяет оптимизировать гнезда циклов по их обработке. Для ускорения расчетов важно уметь равномерно распределять расчеты отдельных блоков между мощностями вычислителя.

Панель конфигурации запуска параллельных расчетов в ESI CFD-FASTRAN для режима MDICE



MDICE-based parallel simulation setup in CFD-FASTRAN

Подготовка расчетной сетки для повышения эффективности параллельных расчетов

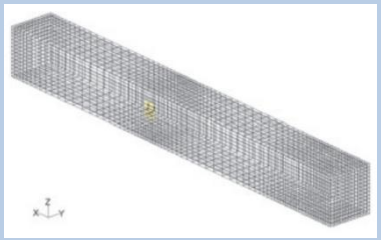
Создание сетки

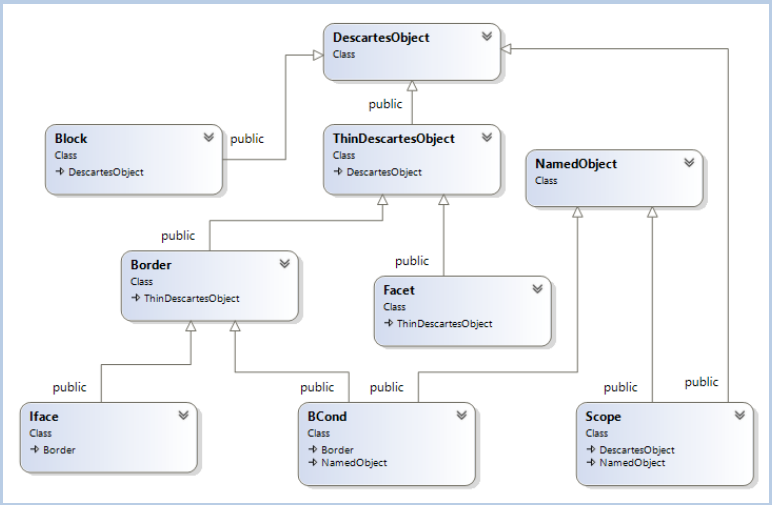
Подготовка сетки (внутреннее представление, импорт/экспорт, морфинг, балансировка)

Счет задачи

CFD-GEOM

ANSYS ICEM CFD





Авторские коды

CFD-FASTRAN

ANSYS Fluent

Для эффективного использования суперкомпьютера требуется подготовка расчетной сетки с учетом всех особенностей конкретной вычислительной системы.

Внутреннее представление блочно-структурированной сетки, совместимое с форматом ESI CFD-GEOM

Multi-Domain Grid File with Blanking

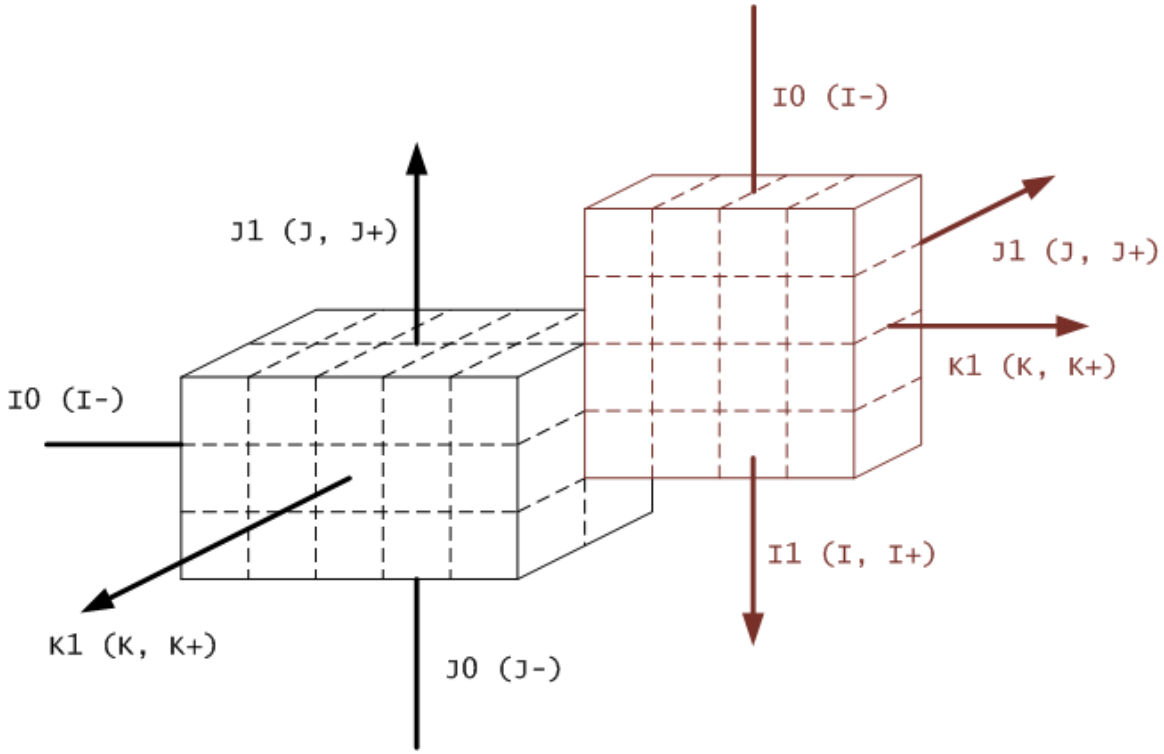
Nz
Ni(1) Nj(1) Nk(1) ... Ni(Nz) Nj(Nz) Nk(Nz)
X1(1,1,1) ... X1(Ni(1),Nj(1),Nk(1)),
Y1(1,1,1) ... Y1(Ni(1),Nj(1),Nk(1)),
Z1(1,1,1) ... Z1(Ni(1),Nj(1),Nk(1)),
IB1(1,1,1)...IB1(Ni(1),Nj(1),Nk(1))
.
.
XNz(1,1,1) ... XNz(Ni(Nz),Nj(Nz),Nk(Nz)),
YNz(1,1,1) ... YNz(Ni(Nz),Nj(Nz),Nk(Nz)),
ZNz(1,1,1) ... ZNz(Ni(Nz),Nj(Nz),Nk(Nz)),
IBNz(1,1,1)...IBNz(Ni(Nz),Nj(Nz),Nk(Nz))

IBn = three dimensional array of i-blanking values for domain n

all other variables = same as for the grid file without blanking

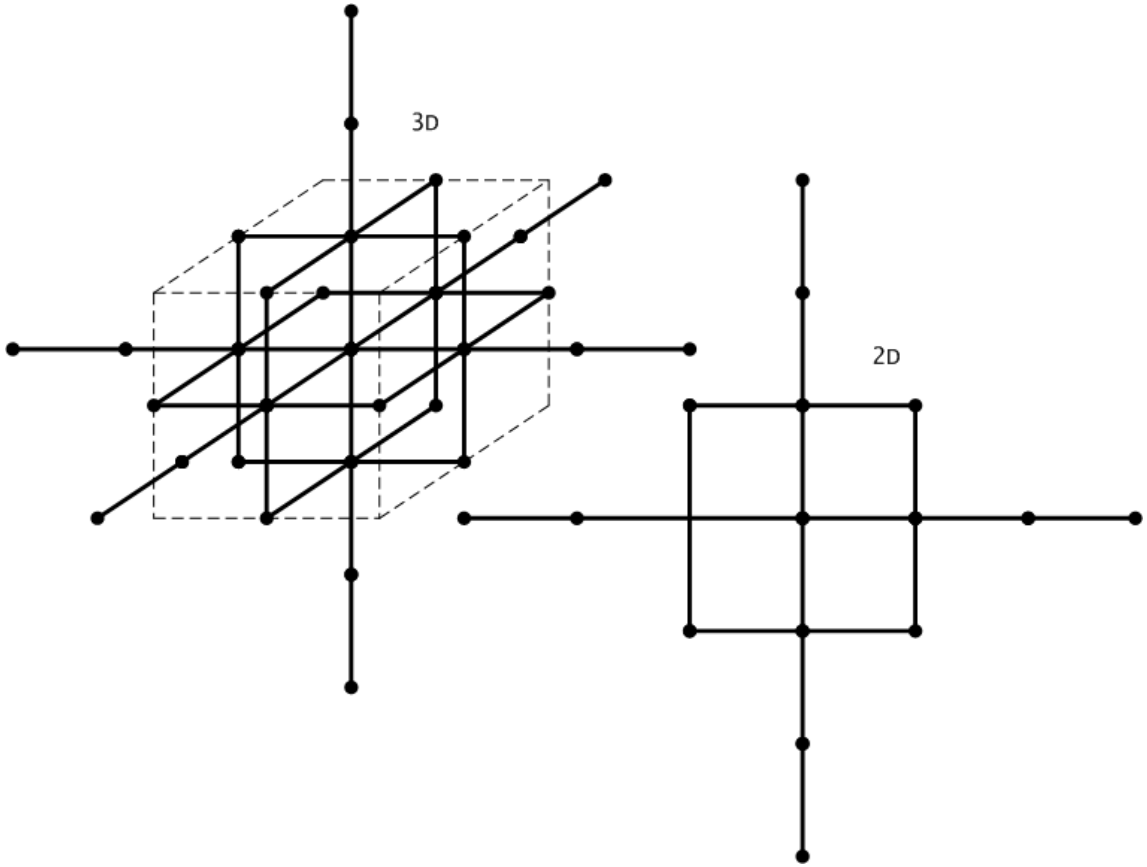
Геометрия блочно-структурированной сетки представлена последовательной записью координат всех узлов каждого блока. Последовательность хранения элементов данных соответствует формату ESI CFD-GEOM.

Система координат, связанная с блоком сетки



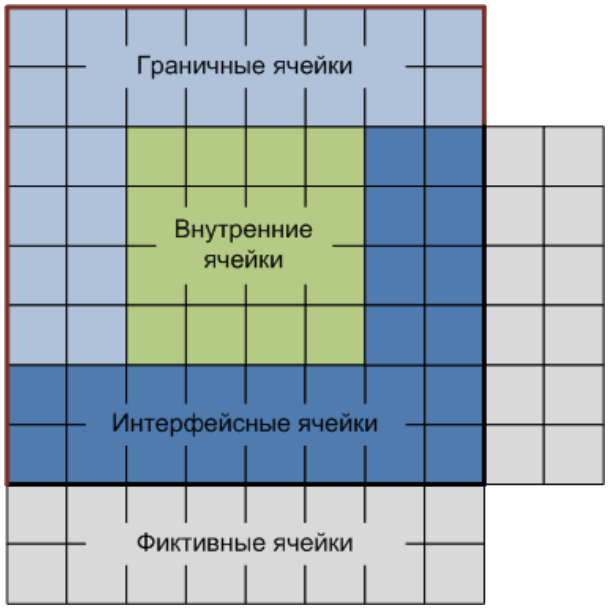
С каждым блоков сетки ассоциирована своя система координат (`I`, `J`, `K`). Каждая координатная ось связывает две противоположные грани блока. Системы координат двух соседних блоков не связаны друг с другом (их оси могут быть произвольно направлены друг относительно друга).

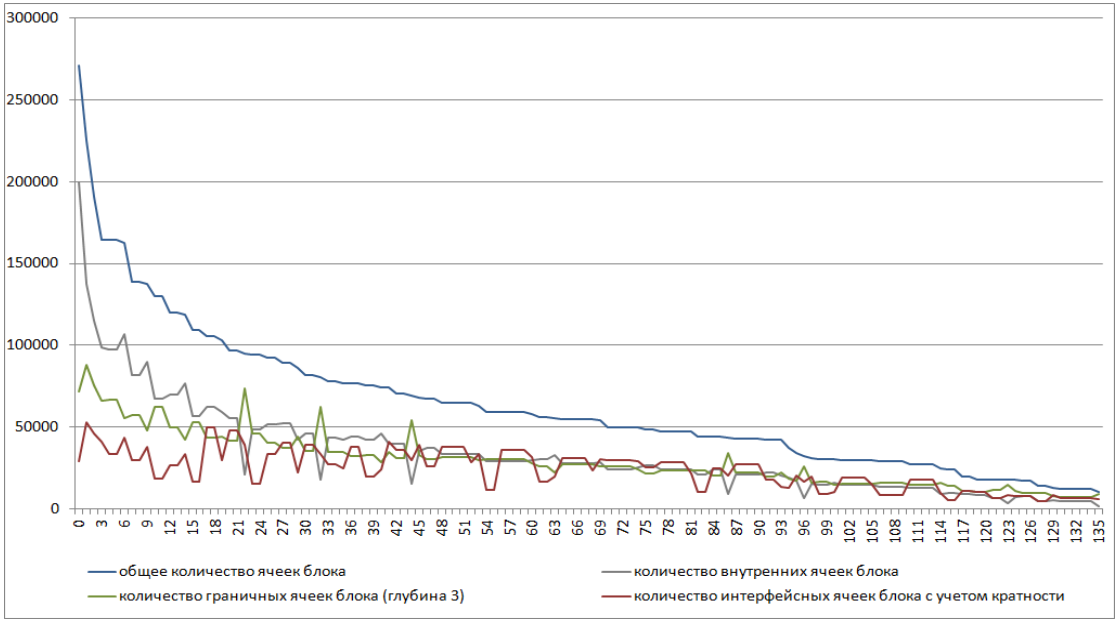
Вычислительные окрестности ячеек сетки



В процессе вычислений во время обработки конкретной ячейки происходит обращение за данными в другие ячейки сетки, в так называемую окрестность ячейки. Размер такой окрестности определяет ширину теневой зоны.

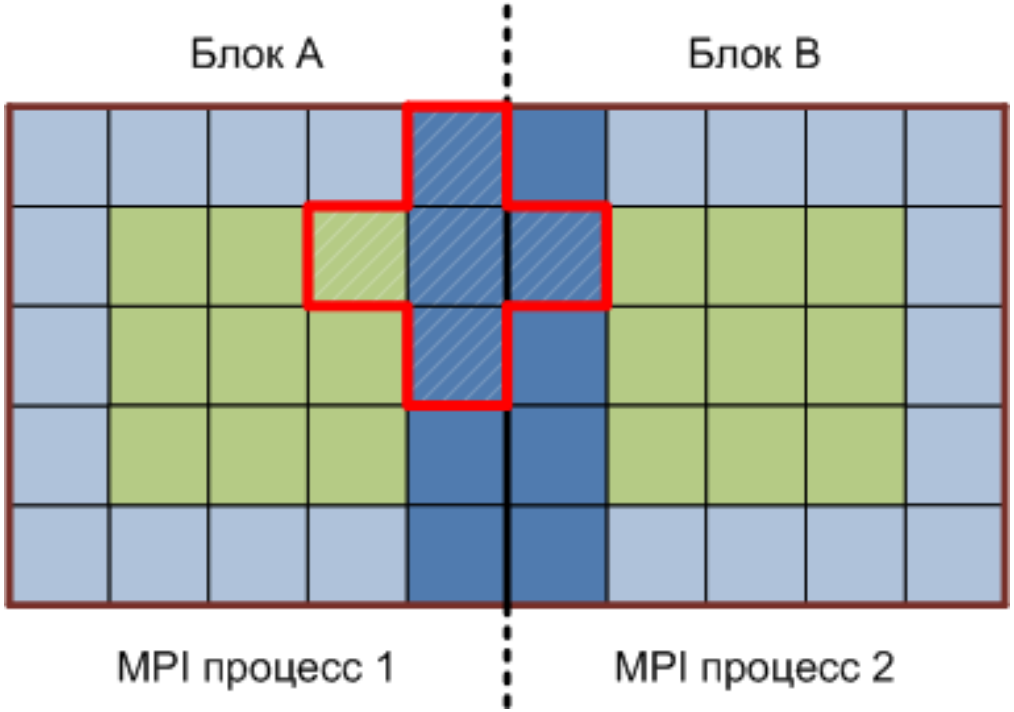
Классификация ячеек блока и характерная статистика по их количеству





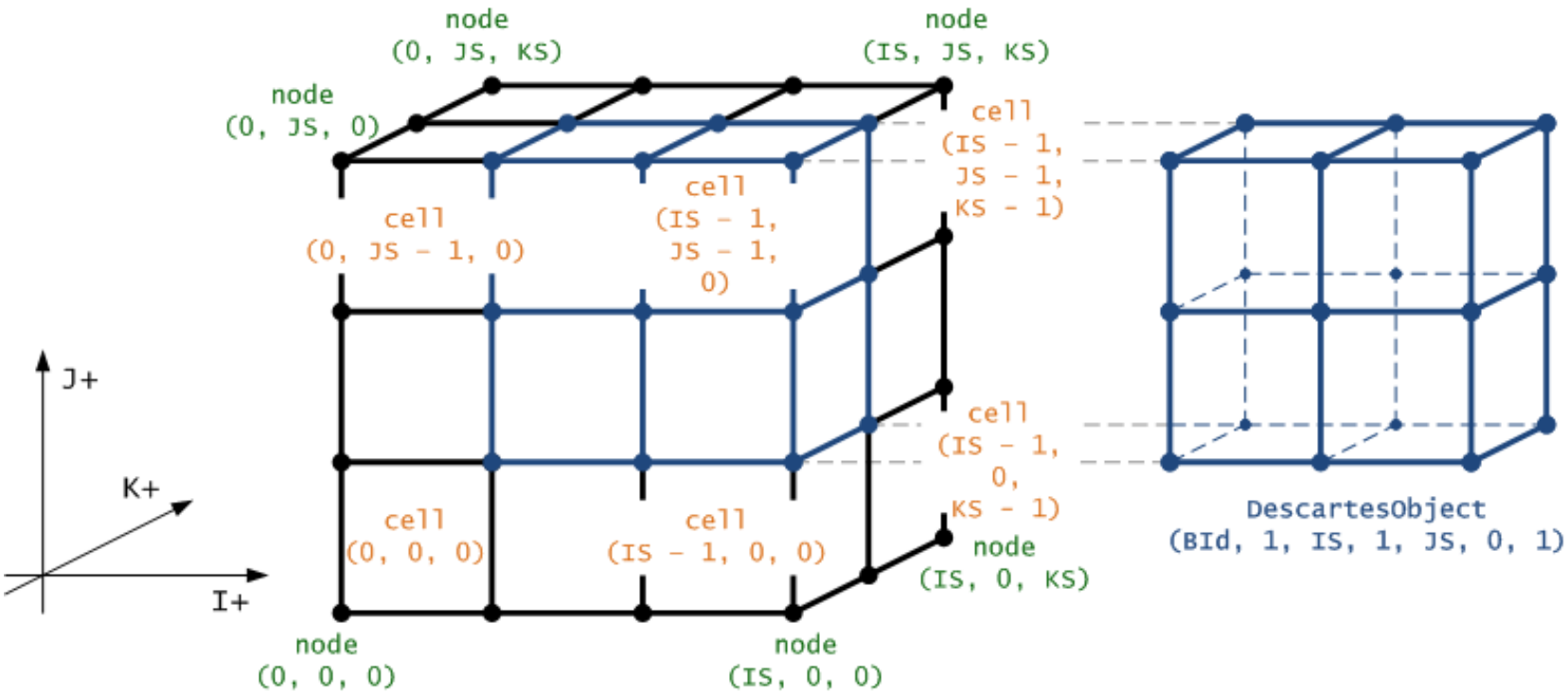
- *Внутренние ячейки* – ячейки, чья вычислительная окрестность целиком лежит в своем блоке.
- *Граничные ячейки* – все остальные ячейки блока.
- *Интерфейсные ячейки* – граничные ячейки, чья вычислительная окрестность частично затрагивает другой блок.
- *Фиктивные ячейки* – копии ячеек соседних блоков, которые нужны для быстрого обращения за данными для интерфейсных ячеек.

Обмен данными между двумя соседними блоками, обрабатывающимися разными процессами



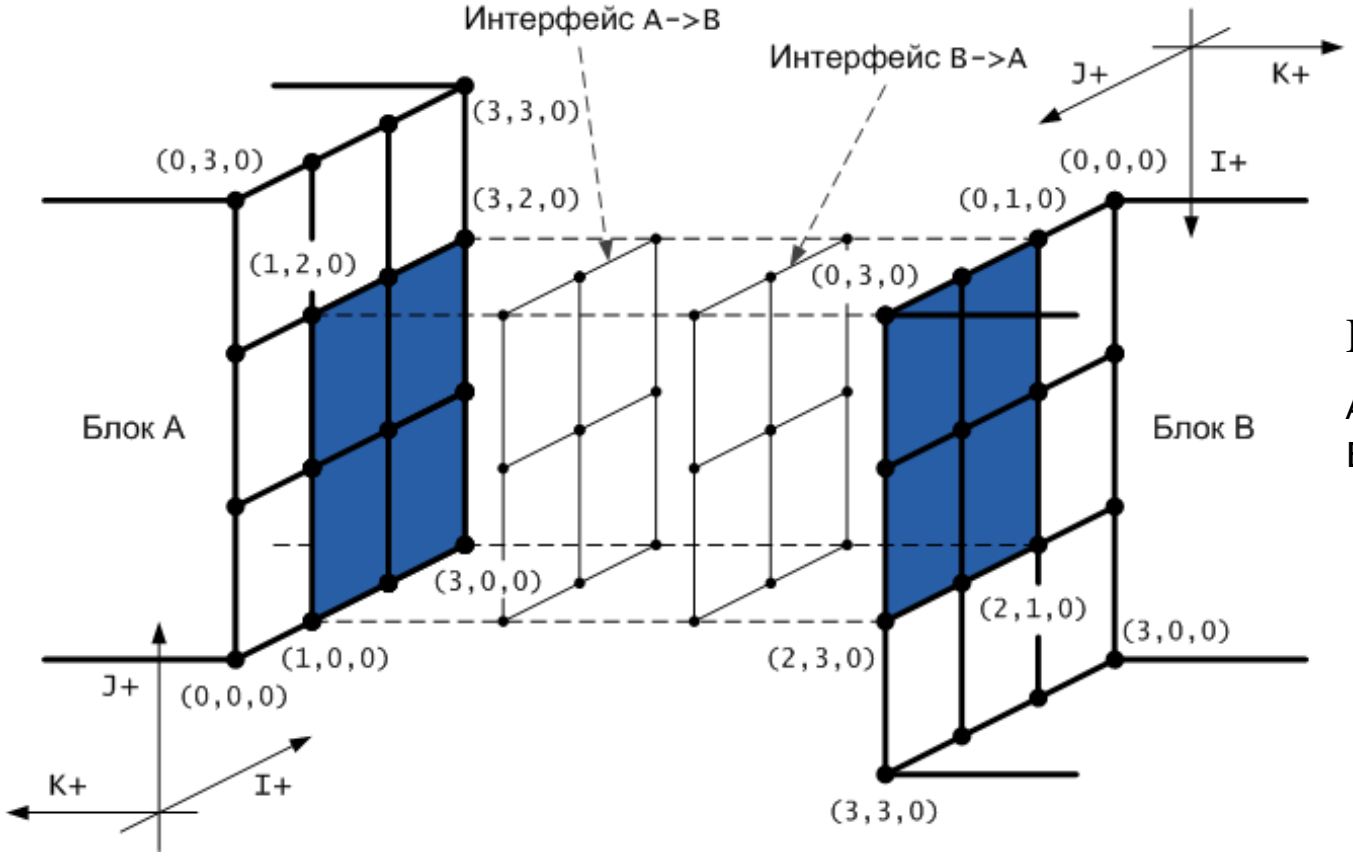
Если во время счета сетка обрабатывается несколькими процессами, то возможна ситуация, когда при обсчете интерфейсной ячейки требуется межпроцессный обмен данными (два соседних блока обрабатываются разными процессами). Такую интерфейсную ячейку будем называть MPI ячейкой.

Узлы блока как координаты его подобластей



Узлы блока представляют собой трехмерный массив точек с заданными координатами по осям `I`, `J`, `K`. Произвольная одномерная, двумерная или трехмерная подобласть блока, являющаяся топологическим параллелепипедом, может быть идентифицирована указанием идентификатора блока и диапазонов номеров узлов по каждому измерению.

Интерфейс – описание касания двух соседних блоков

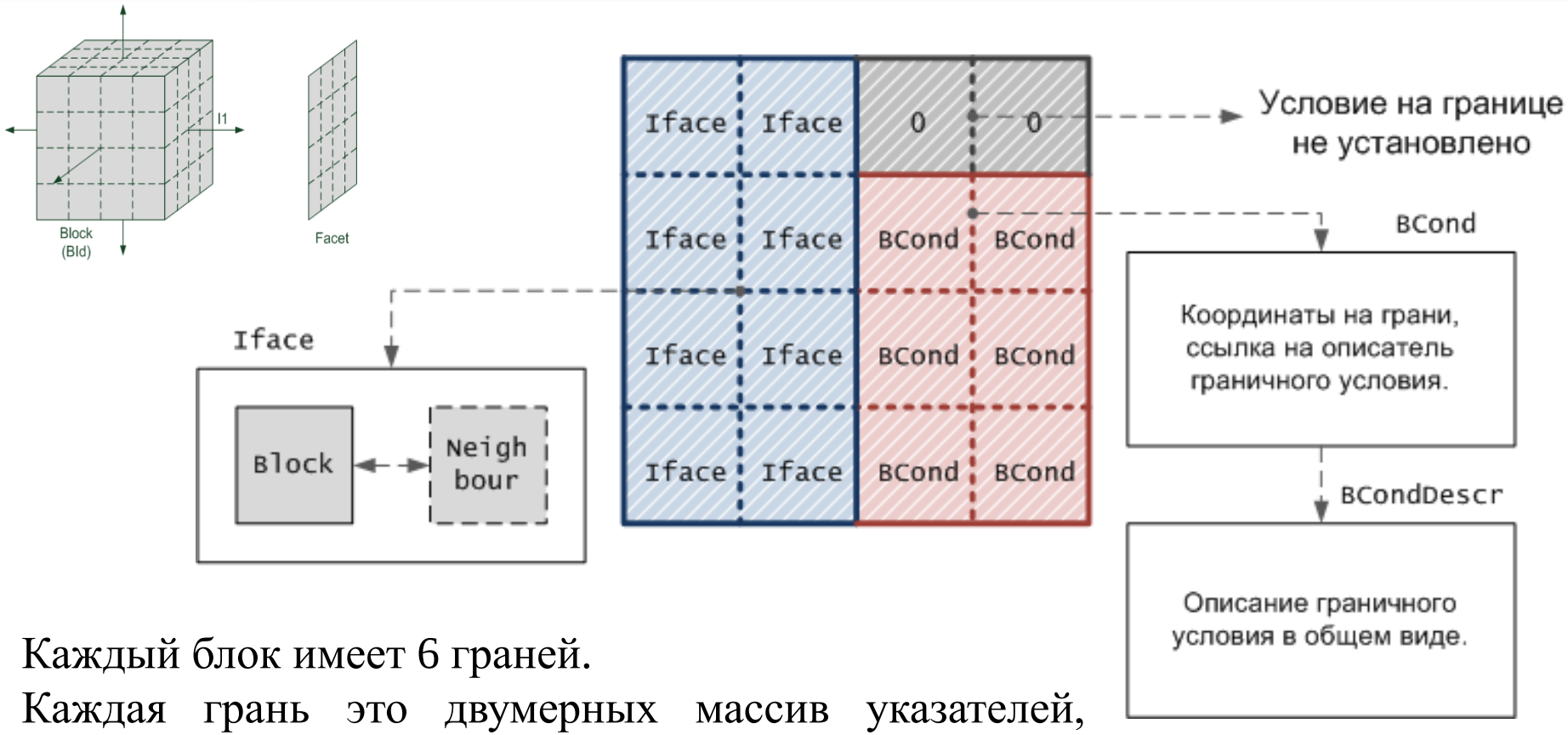


Координаты интерфейсов

```
A->B: 1, 3, 0, 2, 0, 0
B->A: 0, 2, 1, 3, 0, 0
```

Касание двух блоков задается описанием пары направленных друг на друга интерфейсов в координатах номеров узлов соответствующих блоков. Этой информации достаточно для восстановления ориентации систем координат.

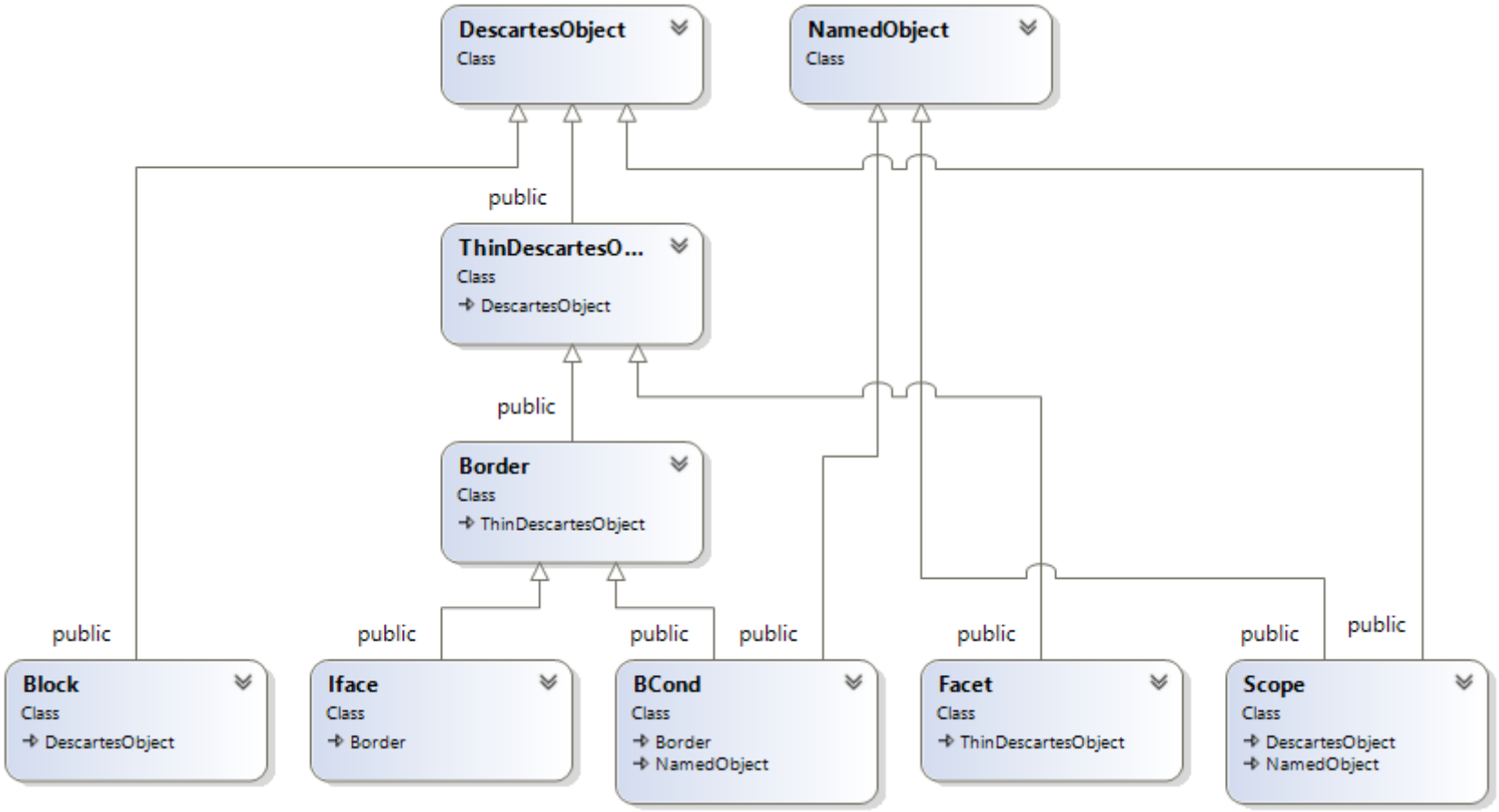
Структура граничных условий



Каждый блок имеет 6 граней.

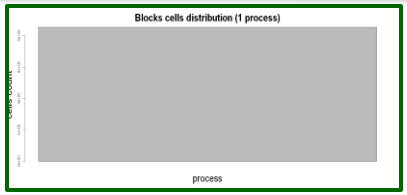
Каждая грань это двумерных массив указателей, каждый из которых указывает либо на граничное условие, либо на интерфейс. Таким образом, можно быстро получить информацию об обработке каждой конкретной ячейки, находящейся на границе блока.

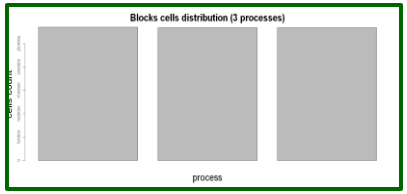
Иерархия основных классов, реализующих блочно-структурированную сетку

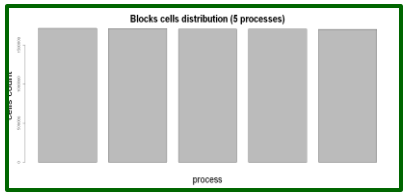


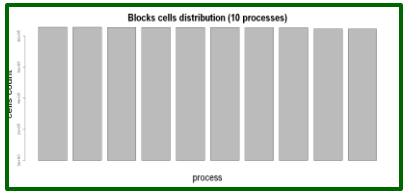
Все основные объекты блочно-структурированной сетки являются декартовыми объектами и обслуживаются единым геометрическим функционалом. Некоторые сложные объекты являются именованными.

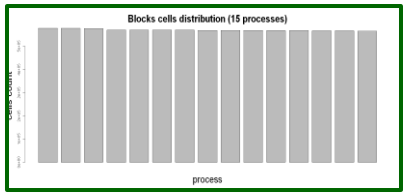
Негативное влияние выраженных крупных блоков на балансировку вычислительной нагрузки

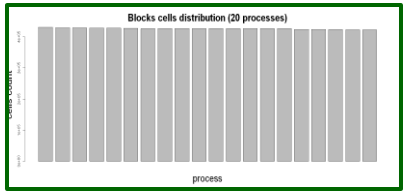


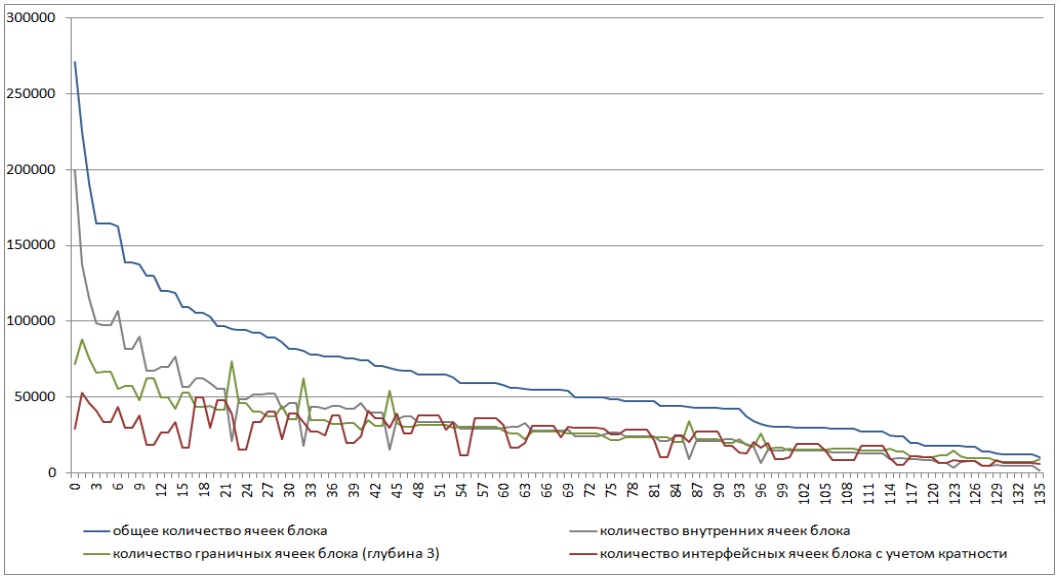




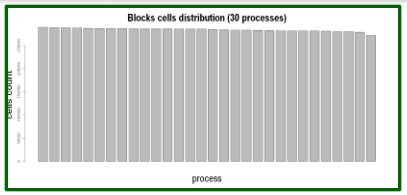


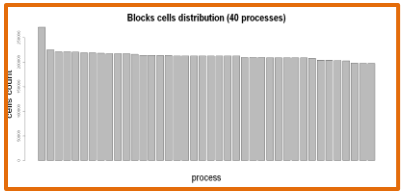


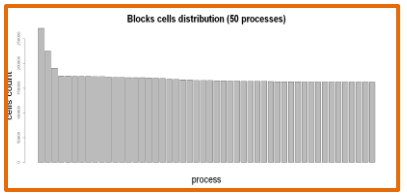


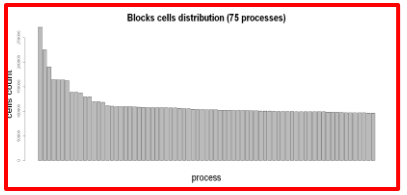


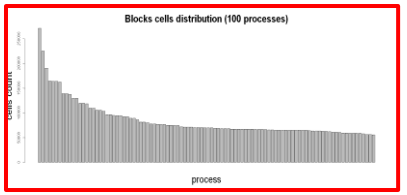
При балансировке нагрузки вычислений на большое количество процессов довольно быстро проявляются выраженные крупные блоки, после чего дальнейшее увеличение количества процессов уже не имеет значения, так как скорость обработки определяется самым загруженным процессом.

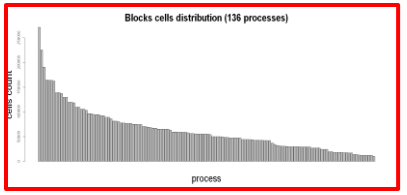




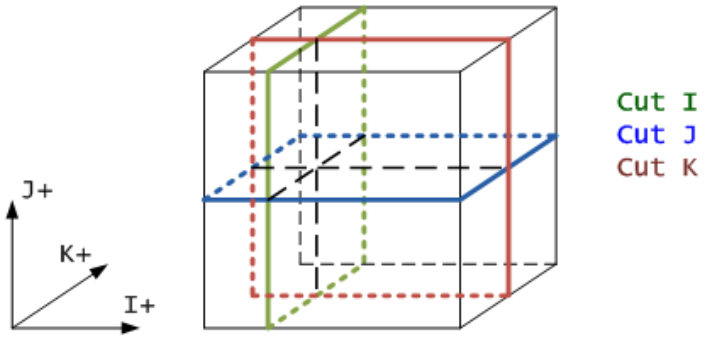






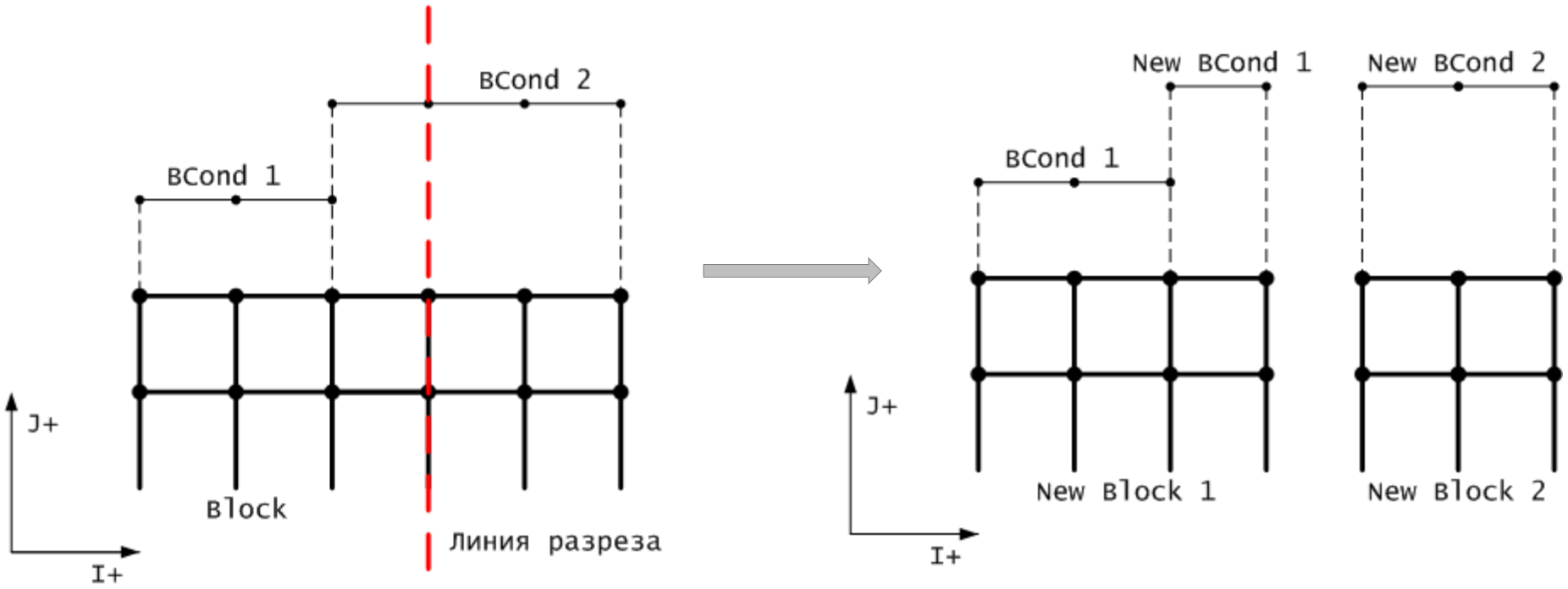


Дробление блока и потенциальное вынужденное дробление связанных элементов

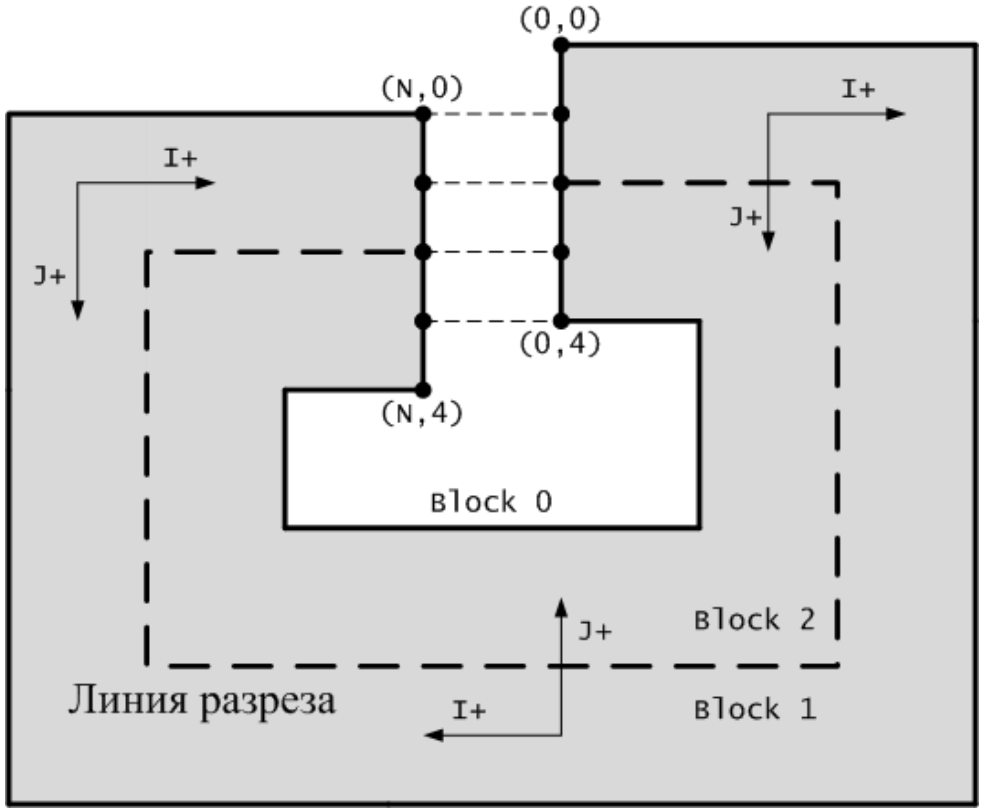


Дробление блока может спровоцировать дробление связанных с ним элементов (областей, интерфейсов, граней, граничных условий).

Функционал дробления осуществляется через класс `DescartesObject`.

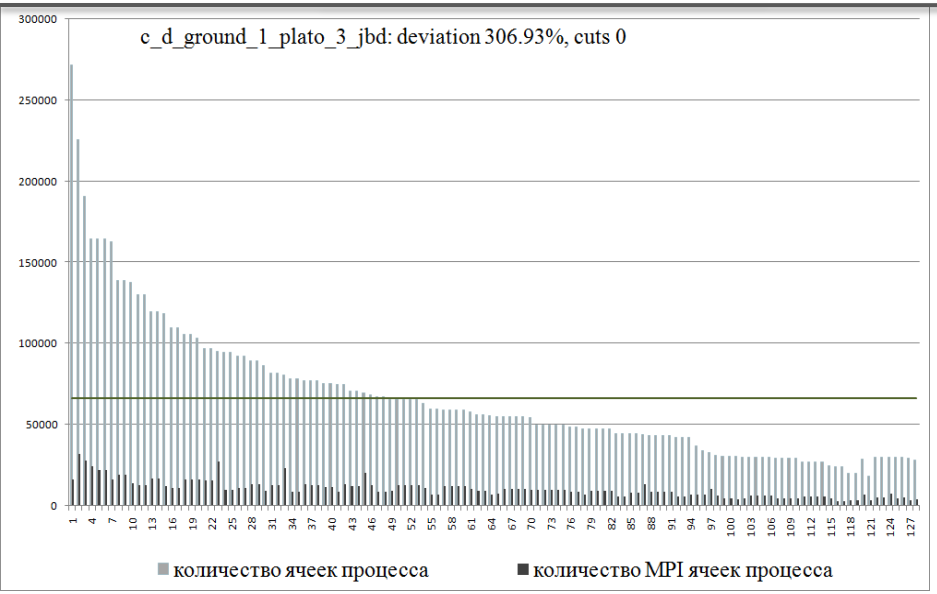


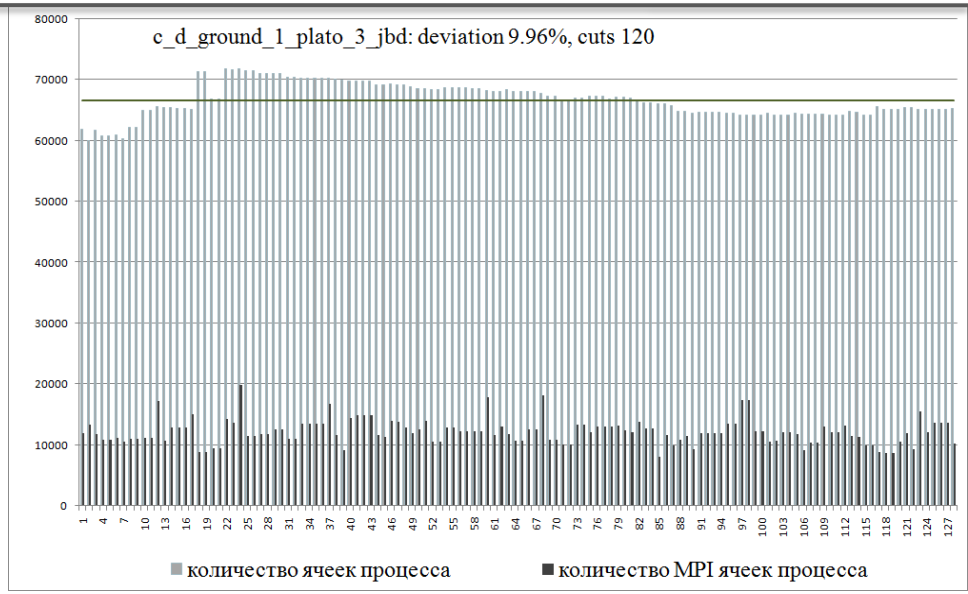
Предельный случай дробления блока

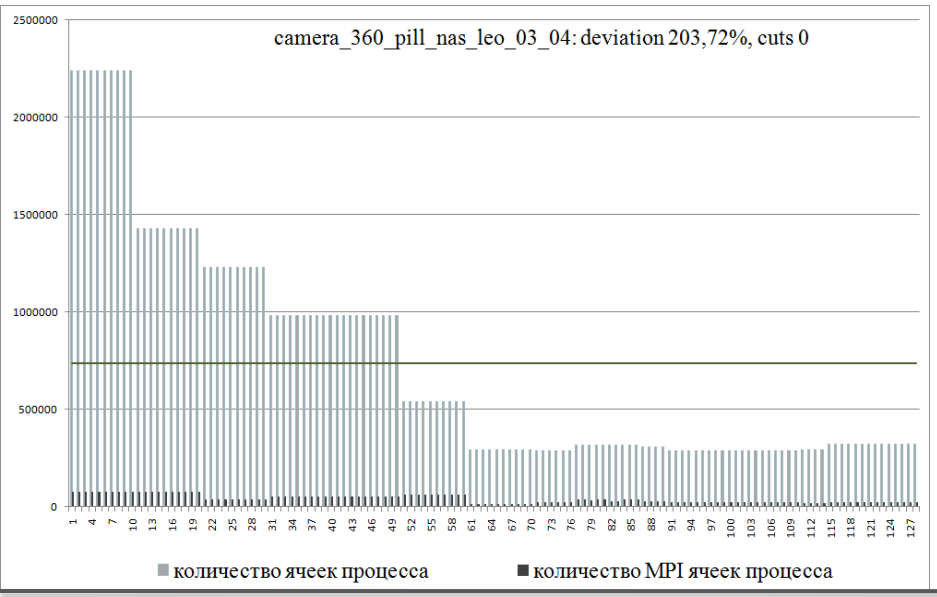


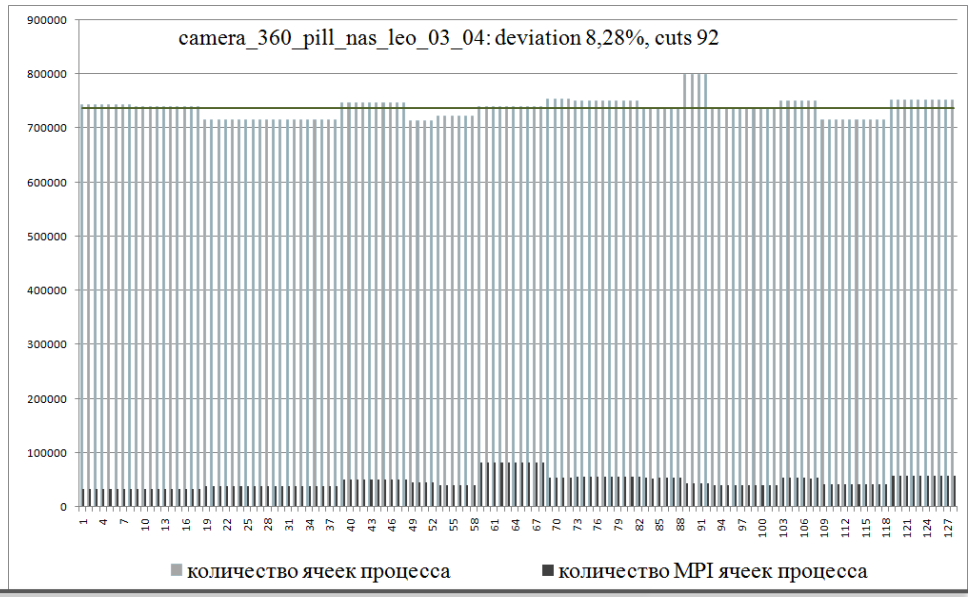
Объектная модель блочно-структурированной сетки позволяет обрабатывать случаи когда дробление блока провоцирует повторное дробление этого же блока по другому разрезу или даже по другому направлению. Обычно генераторы сетки запрещают создание таких блоков.

Эффект от применения дробления сетки при балансировке нагрузки на 128 вычислительных процессов

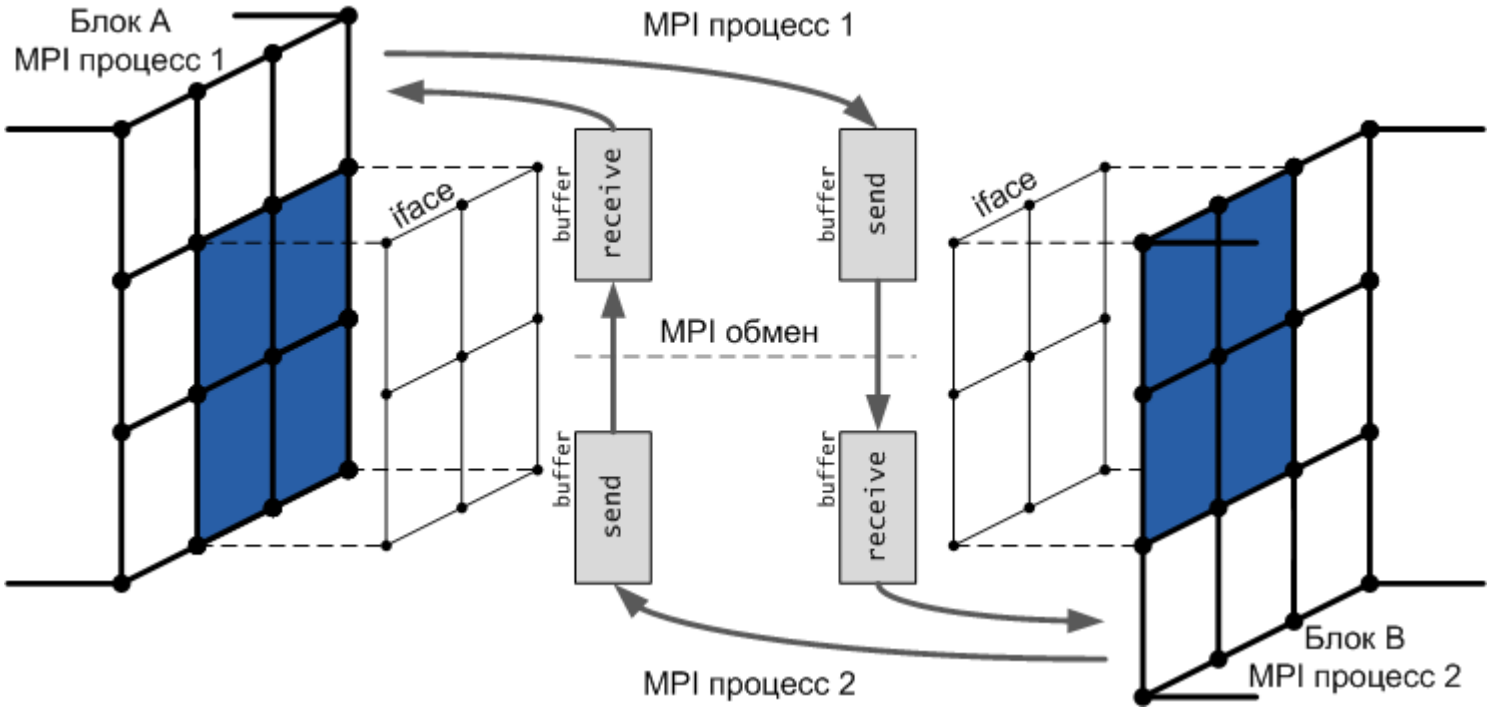








Организация обмена данными между разными процессами MPI



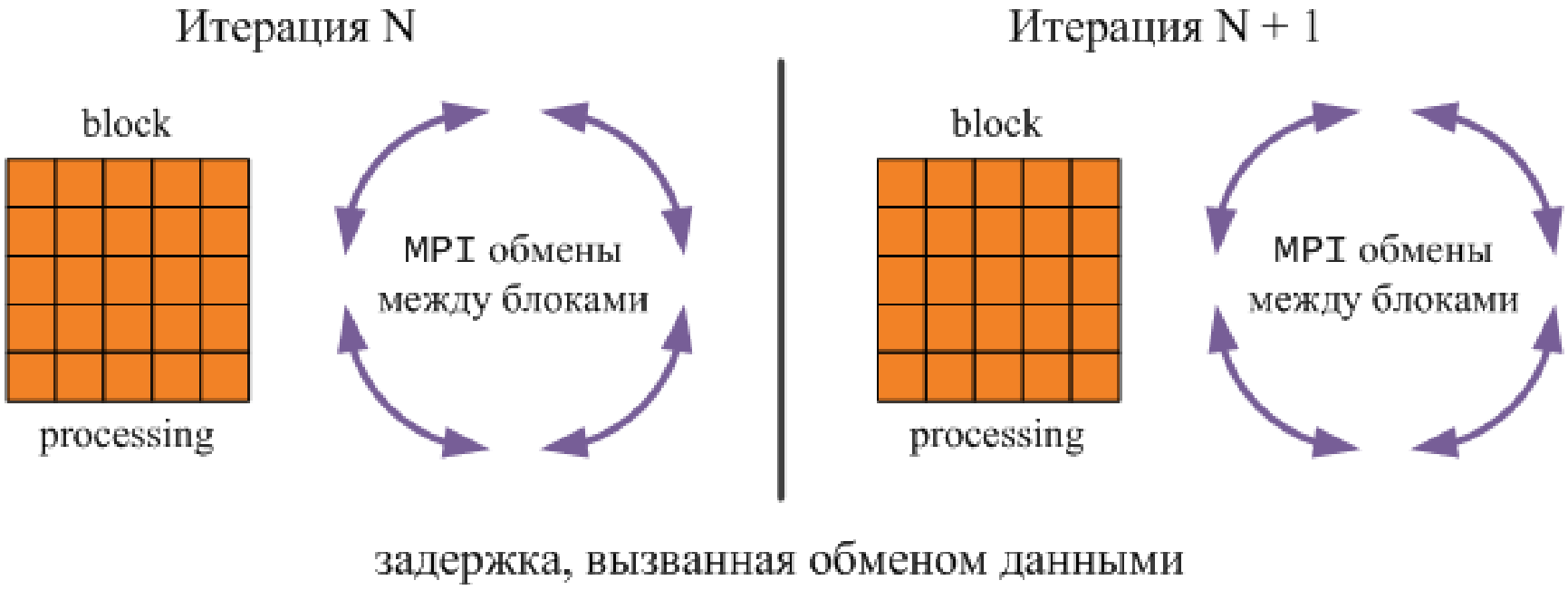
- В обмене данными участвуют интерфейсные ячейки блоков.
- Весь обмен данными на одной итерации счета происходит одновременно с использованием функций асинхронной передачи данных.
- Обмен данными происходит через специальные буферы, являющиеся атрибутами интерфейсов (буфер является принимающим или отправляющим в зависимости от процесса и активности в нем конкретного блока).

Схема MPI обменов (цикл по всем интерфейсам)

```
void StructuredGrid::IfacesDataExchange()
{
    <подготовка массива reqs - запросов>
    <подготовка массива stats - статусов запросов>

    int reqs_count = 0;
    int i = 0;

    // Обработка всех интерфейсов.
    while (i < IfacesCount())
    {
        Iface *p = Ifaces->Get(i);

        if (p->IsBActive())
        {
            if (p->IsNActive())
            {
                // Оба блока интерфейса активны.
                <обмен данными в рамках одного процесса>
                i += 2;
            }
            else
            {
                // Собственный блок активен, соседний блок не активен.
                MPI::IRecvDoubles(p->BufferDoubles(),
                                  p->BufferDoublesCount(),
                                  p->NB()->MPIRank(),
                                  p->Id(),
                                  reqs->ReqP(reqs_count++));

                // Продвижение номера интерфейса.
                i++;
            }
        }
        else
        {
            if (p->IsNActive())
            {
                // Соседний блок активен, собственный блок не активен.
                MPI::ISendDoubles(p->BufferDoubles(),
                                  p->BufferDoublesCount(),
                                  p->B()->MPIRank(),
                                  p->Id(),
                                  reqs->ReqP(reqs_count++));

                // Продвижение номера интерфейса.
                i++;
            }
            else
            {
                // Оба блока не активны, ничего делать не надо.
                i += 2;
            }
        }
    }

    // Ждем завершения всех обменов.
    MPI::WaitAll(reqs_count, reqs, stats);
}
```

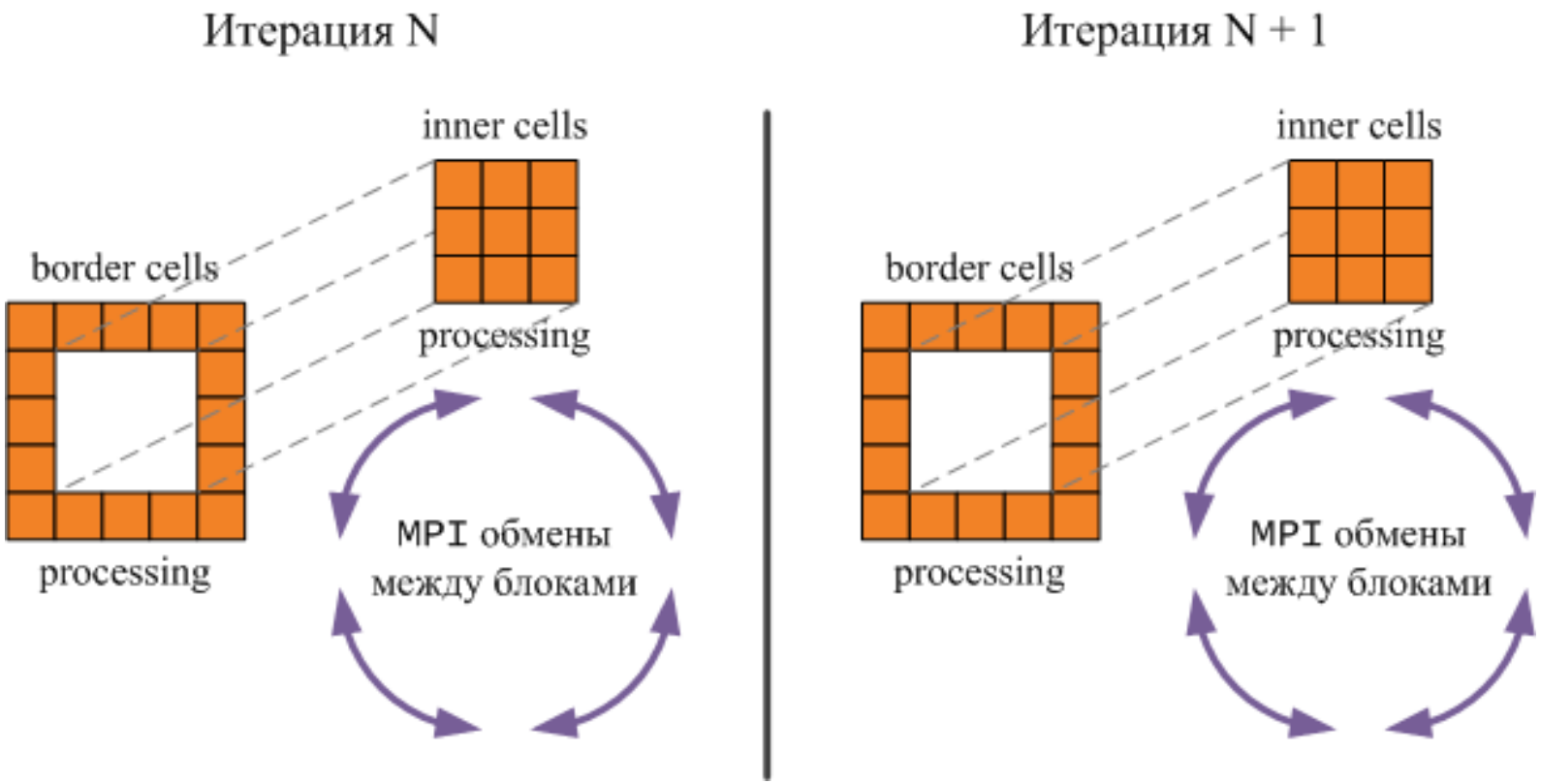
Обмен данными инициируется сразу по всем интерфейсам. Обмен с помощью функций MPI выполняется только если в текущем процессе рассматриваемый блок активен, а его сосед – нет, и наоборот

Последовательность вычислений и MPI обменов на итерации



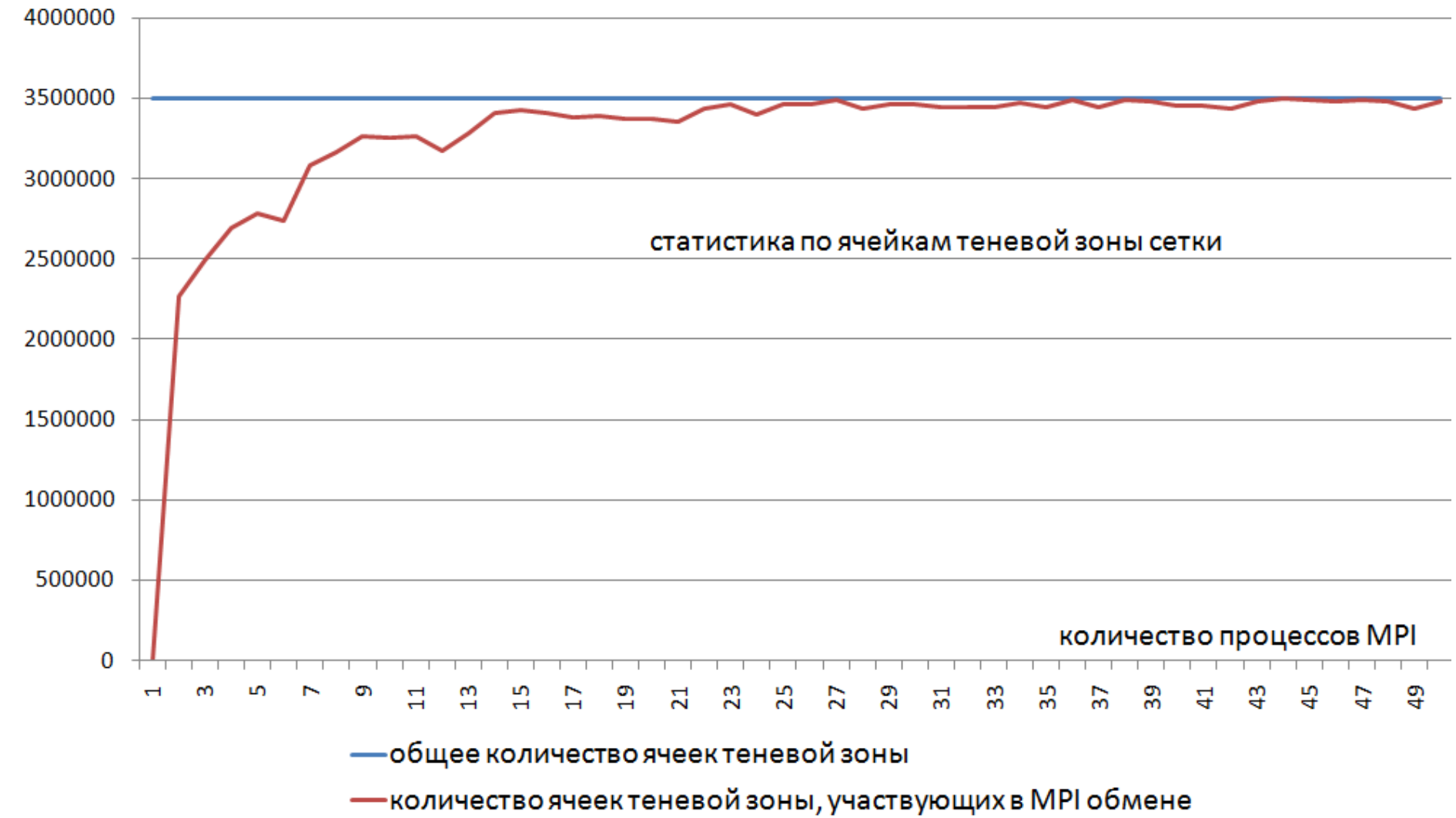
После обсчета всех ячеек сетки на итерации N необходимо произвести обмен данными между MPI ячейками. До окончания обмена нельзя выполнять вычисления следующей итерации.

Разделение вычислений на обработку внутренних и граничных ячеек блока

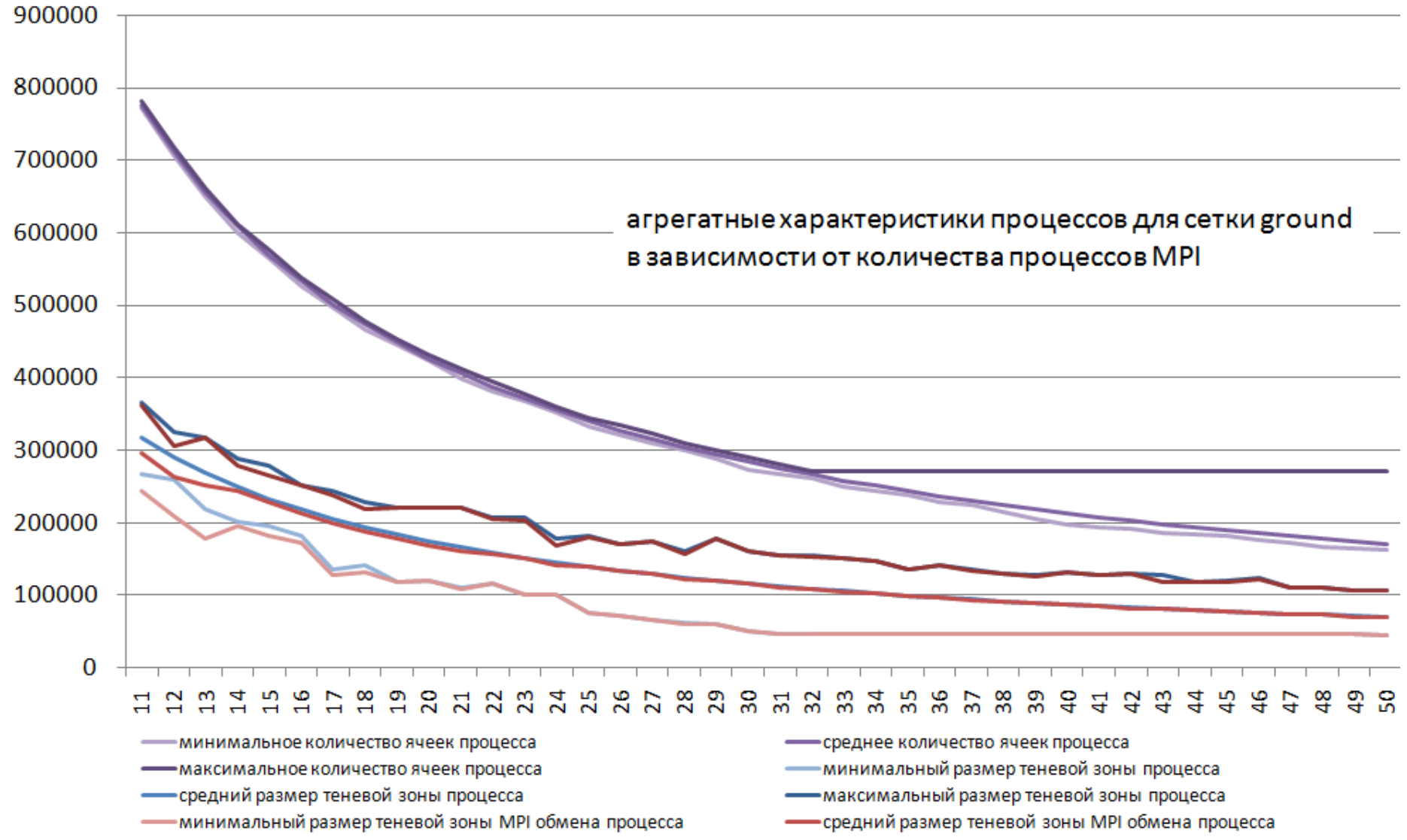


Разделение обработки ячеек блока на обработку внутренних и граничных ячеек позволяет экранировать MPI обмены данными, так как эти обмены не затрагивают внутренние ячейки.

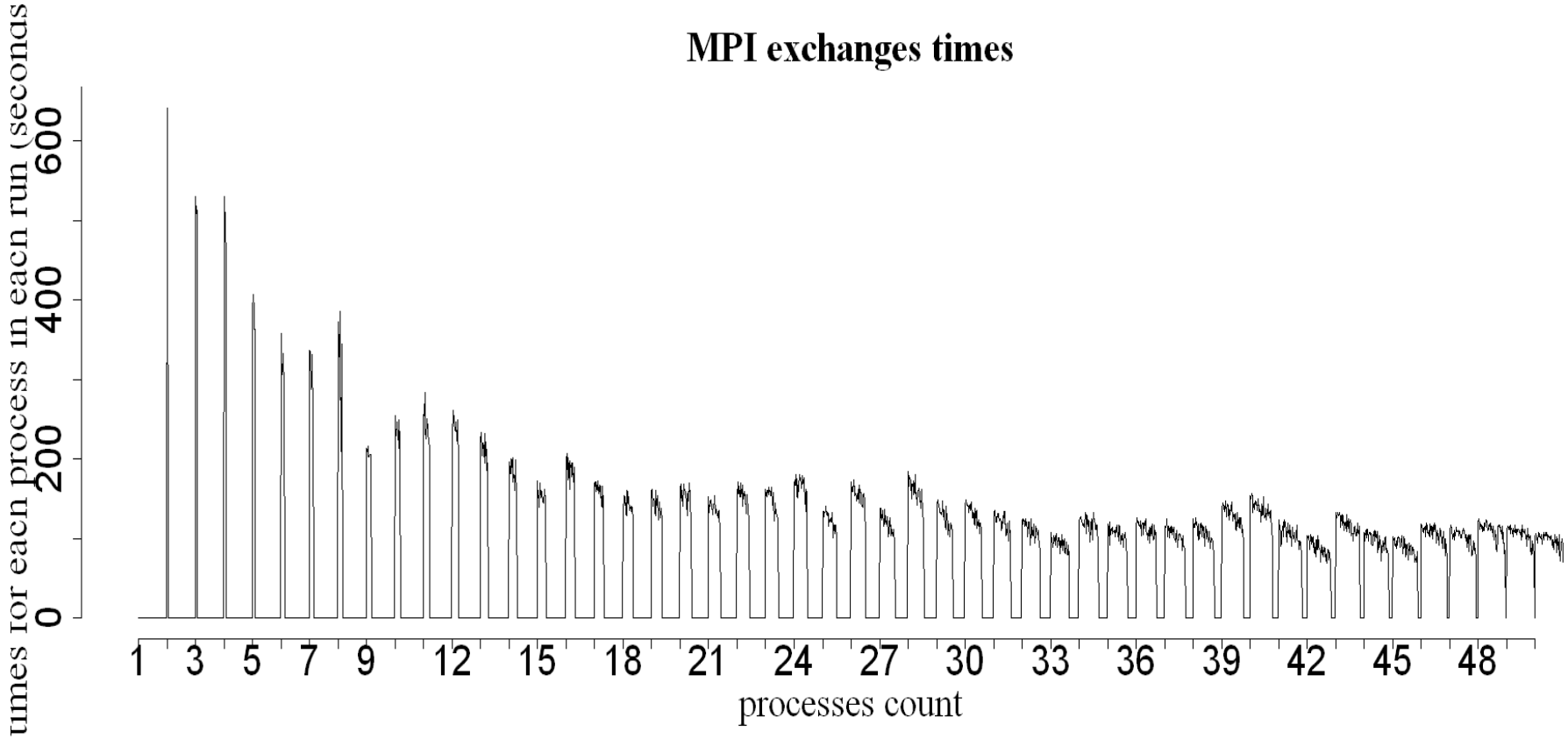
Зависимость размера теневой зоны сетки от степени распараллеливания вычислений



Зависимость агрегатных характеристик теневой зоны процесса от степени распараллеливания вычислений



Зависимость времени MPI обменов от степени распараллеливания вычислений



Результаты замеров показали, что при увеличении степени распараллеливания размер теневой зоны быстро возрастает до своего предельного значения, однако время MPI обменов убывает, что согласуется с агрегатными характеристиками теневой зоны единичного процесса.

Спасибо за внимание.