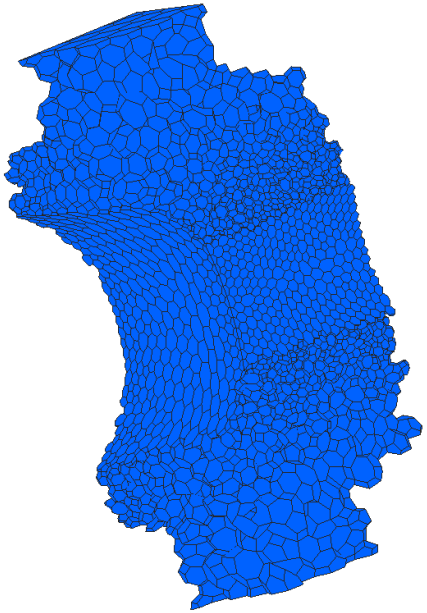
Параллельный топологический алгоритм корректировки нормалей неструктурированной многогранной расчетной сетки на Intel Xeon Phi

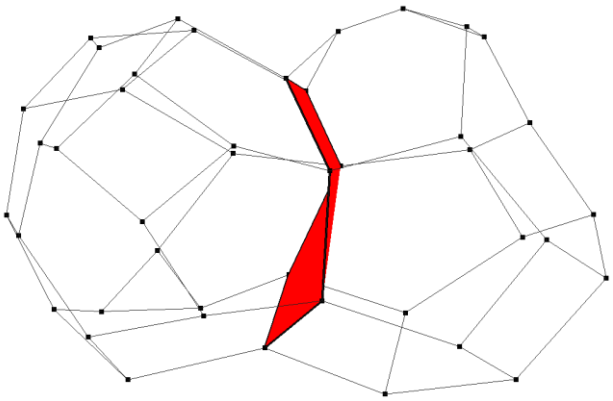
Семенов А.С., Мукосей А.В., Головина Е.А.
DISLab / ОАО «НИЦЭВТ»

5 марта 2015

Научно-технический семинар GraphHPC-2015

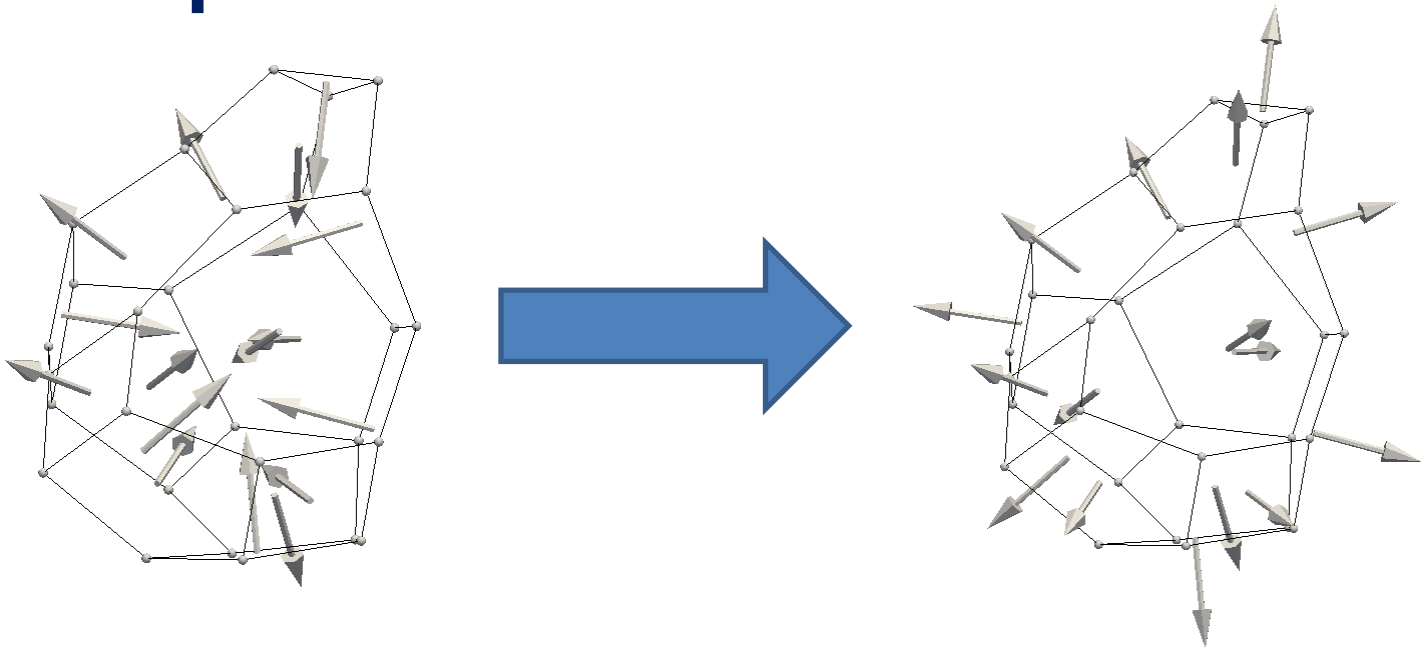
Неструктурированная многогранная расчетная сетка





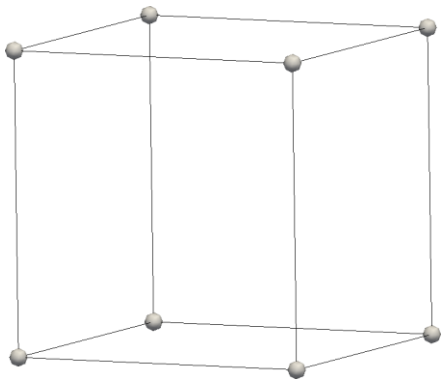
- Использование неструктурированных расчетных сеток на инженерных газодинамических (CFD) кодов для конечно-объемных методик
- Грань – <u>произвольный многоугольник</u> (невыпуклый и неплоский)
- **«+»** Применимы при сложной геометрии
- **«+»** Автоматическое построение расчетной сетки для сложных геометрий
  (в отличии от трудоемкого ручного построения для блочно-структурированных сеток)

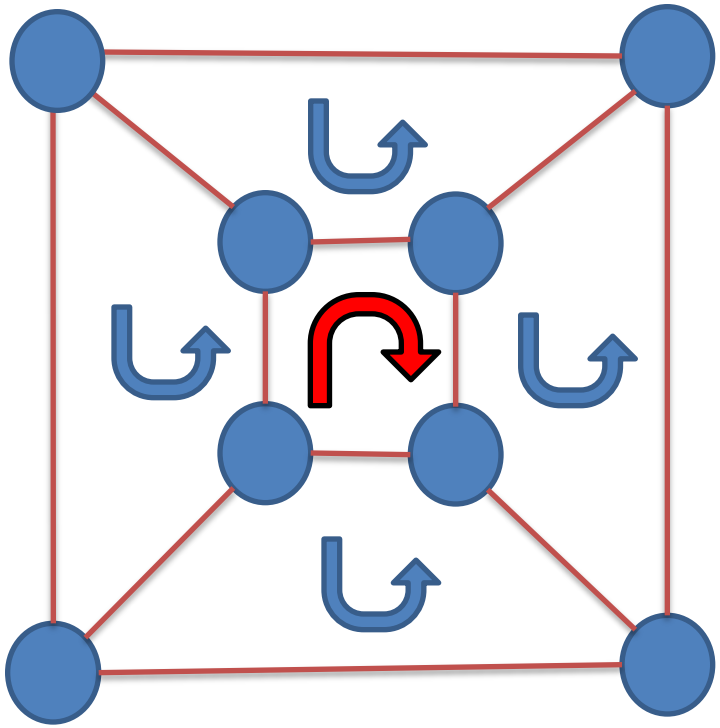
Проблема корректировки нормалей, геометрический и топологический метод



- Нормаль грани определяется узлами грани, направление по “правилу правой руки” (буравчика)
- В газодинамических конечно-объемных кодах необходимо чтобы нормаль грани была направлена **<u>из ячейки</u>**
- Некоторые генераторы неструктурированных сеток не гарантируют правильное направление нормали грани
- Геометрический метод корректировки направления нормали труден в реализации для невыпуклых ячеек

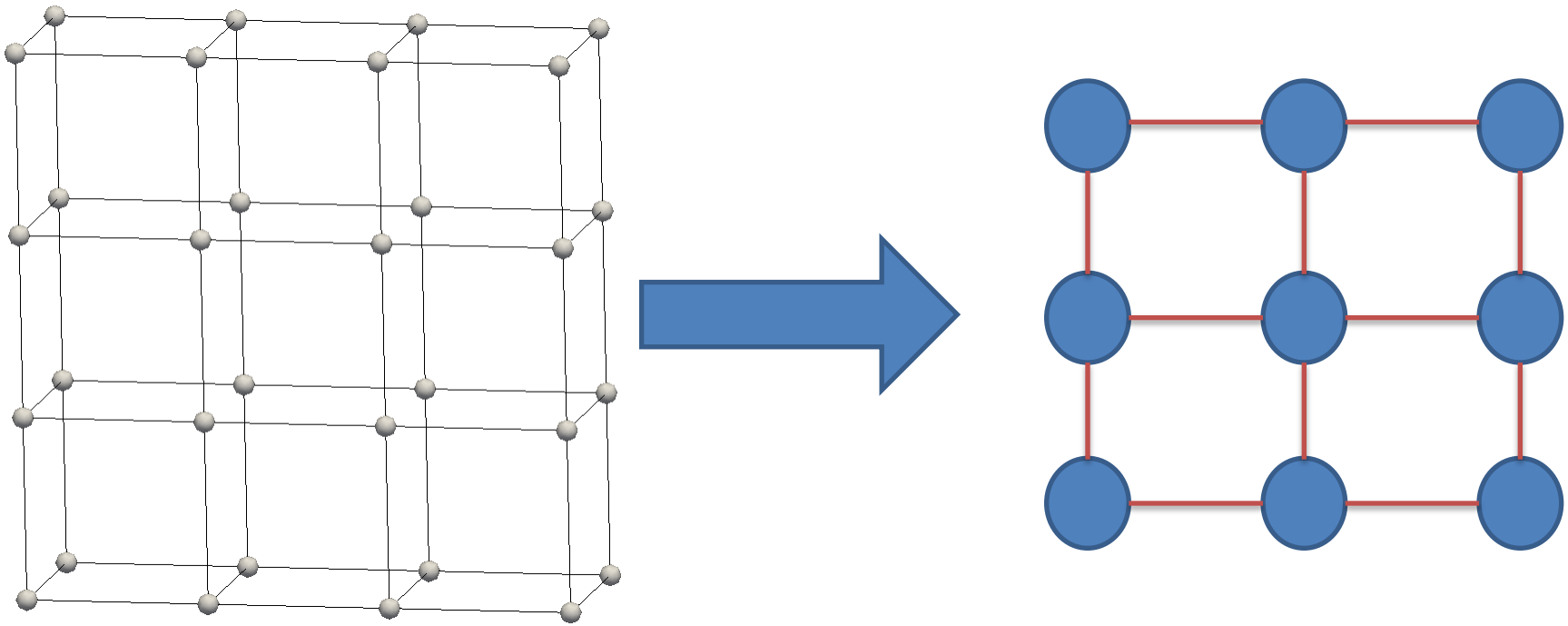
Алгоритм “ПротивоХод” корректировки нормалей в многогранной ячейке Функция correctCell





- Узлы каждого ребра должны перечисляться в двух смежных гранях в разном (противоположном) порядке
- Нужно иметь одну «начальную» правильную грань, по которой можно восстановить все остальные грани

Корректировка нормалей расчетной сетки. Обход графа в ширину (BFS)



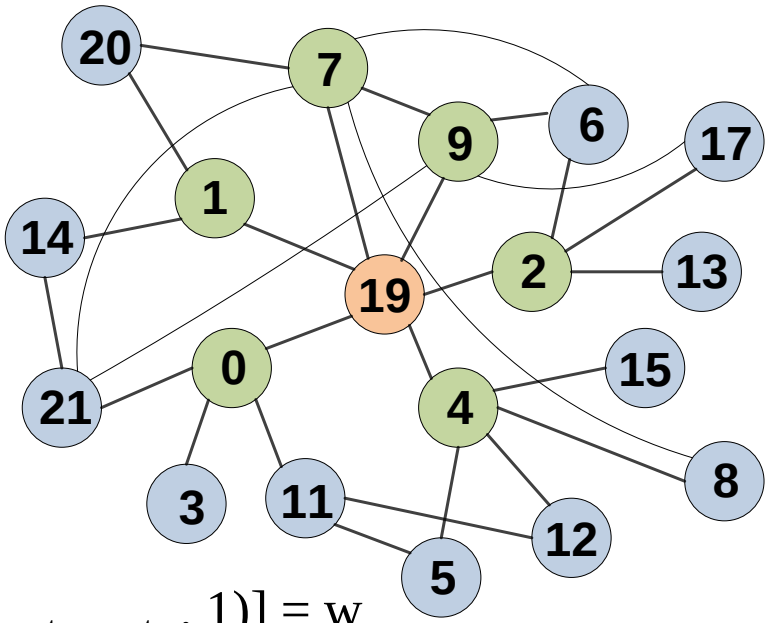
- Расчетная сетка => граф
- Ячейка => вершина
- Внутренняя грань => ребро

Мотивация распараллеливания

- Важно время генерации больших расчетных сеток
- В некоторых методиках расчетная сетка корректируется/перестраивается на каждом шаге
- Необходимы параллельные многопоточные алгоритмы (например, MPI+OpenMP)

Алгоритм Противоход, версия atom

```
Q_counter = 1
Q[0] = root
marked[root] = 1
while Q_counter > 0
    Q_next_counter = 0
    #pragma omp parallel for
    for all vertex ∈ Q do
        for all w: (vertex, w) ∈ E do
            if marked[w] == 0 then
                Q_next[__sync_fetch_and_add(Q_next_counter, 1)] = w
                marked[w] = 1
                correctCell_atom(w)
            endif
        end for
    end for
    swap(Q, Q_next)  // обмен Q и Q_next
end while
```



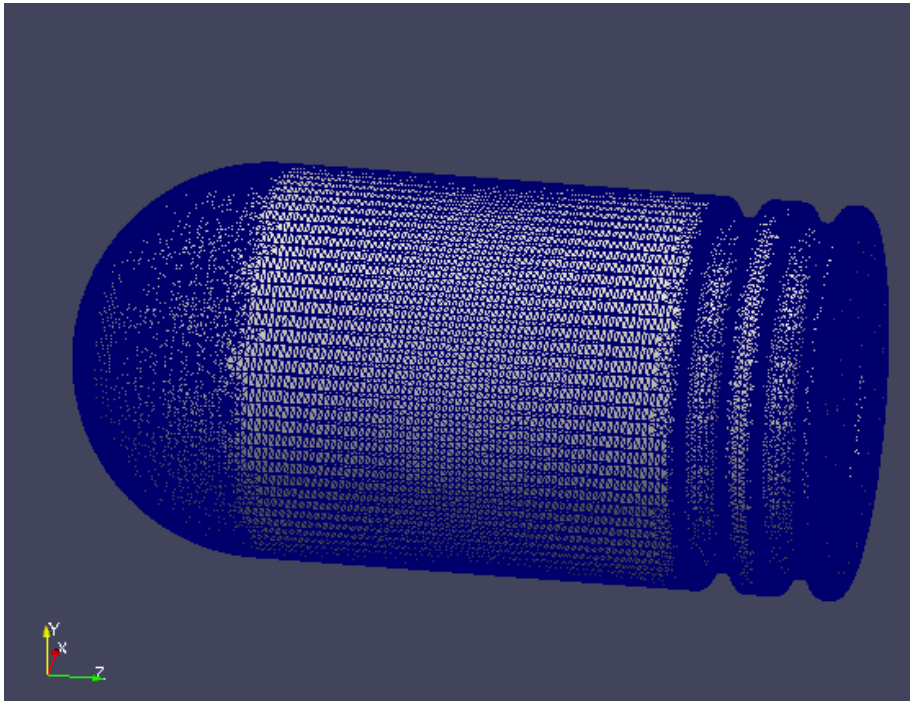
Массив Faces, элементы {0, 1}

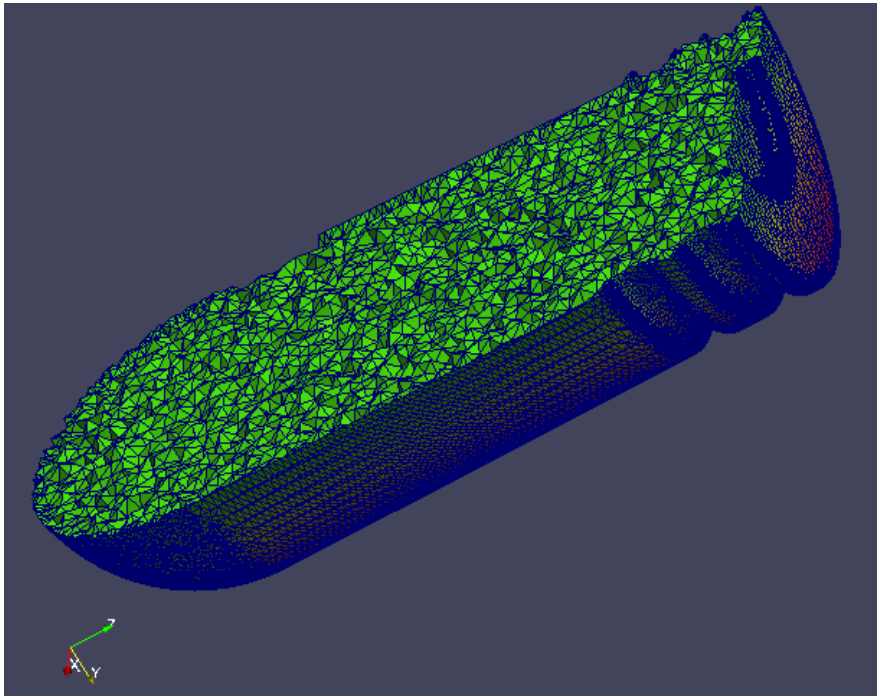
Характеристики используемых процессоров

	Sandy Bridge-EP	Phi-5110P
Название модели	Xeon E5-2660	Xeon Phi 5110P
Частота	2.2 ГГц	1.05 ГГц
Количество сокетов	1	1
Количество ядер	8	60
Количество аппаратных тредов в ядре	2	4
Размер памяти кэшей данных (* - на одно ядро)	64 КБ* / 2 МБ* / 20 МБ	32 КБ* / 512 КБ*
Объем установленной памяти	32 ГБ	8 ГБ
Тип используемой памяти	DDR3	GDDR5
Пиковая пропускная способность памяти	51 ГБ/с	352 ГБ/с
Пропускная способность чтения из памяти, ГБ/с; Последовательный / случайный доступ	42 / 3.3	183 / 3.8
Задержка обращения в память	~200 тактов	~300 тактов

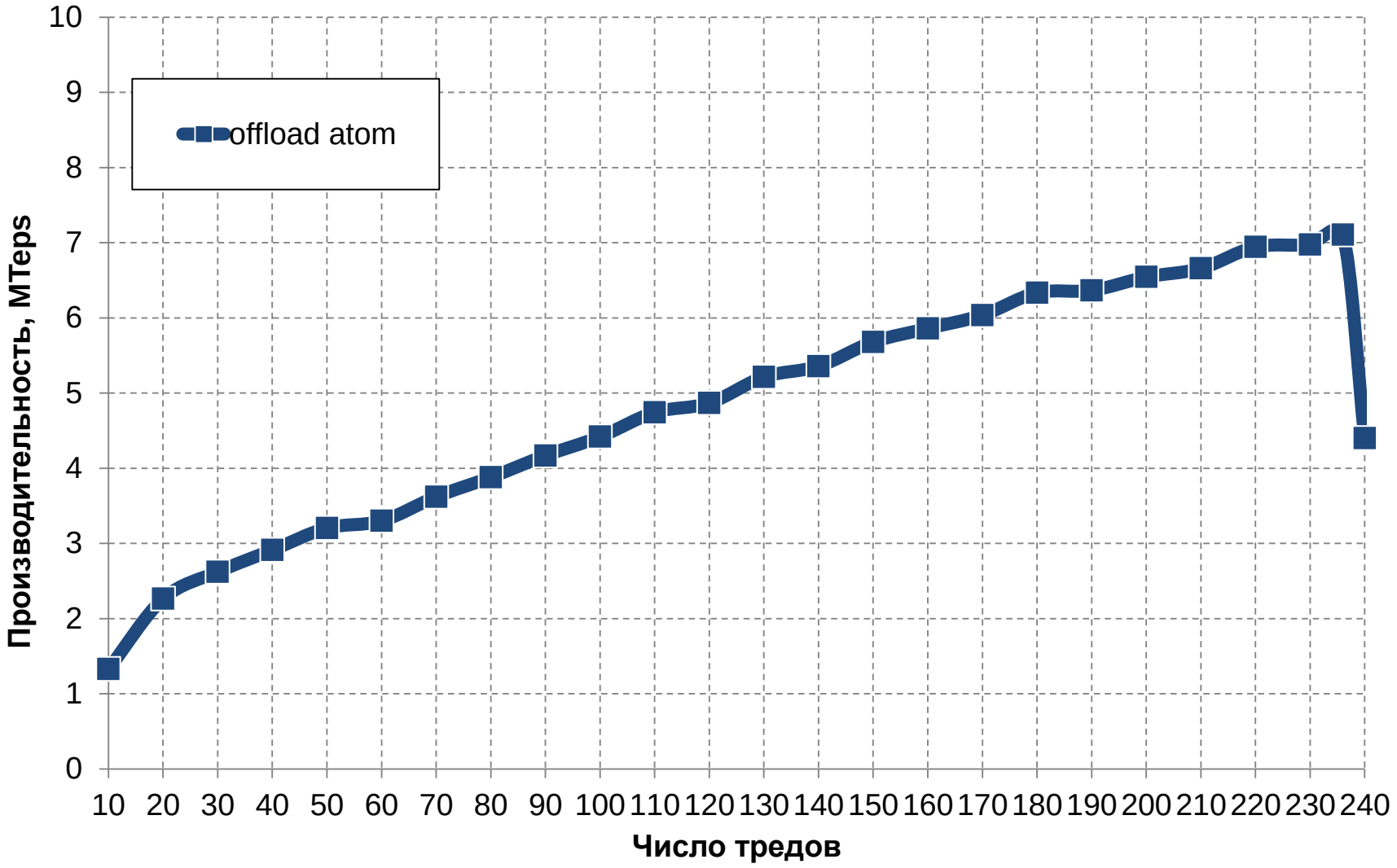
Используемые сетки: TetGen

- Открытый генератор сеток **http://wias-berlin.de/software/tetgen**
- Трехмерные ячейки в форме тетраэдров
- Размеры от 50 МБ до 4.7 ГБ



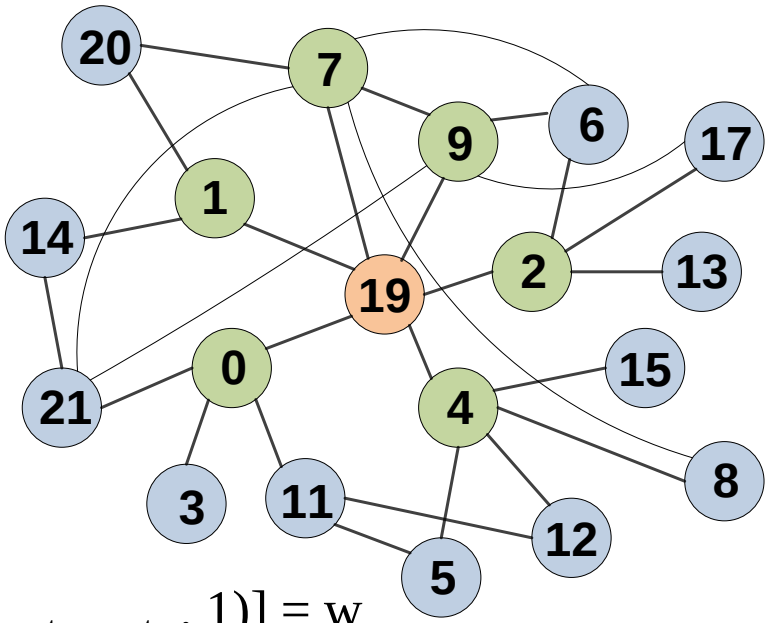


Производительность алгоритма Противоход на Intel Xeon Phi 5110P (сетка - 3.5 ГБ)



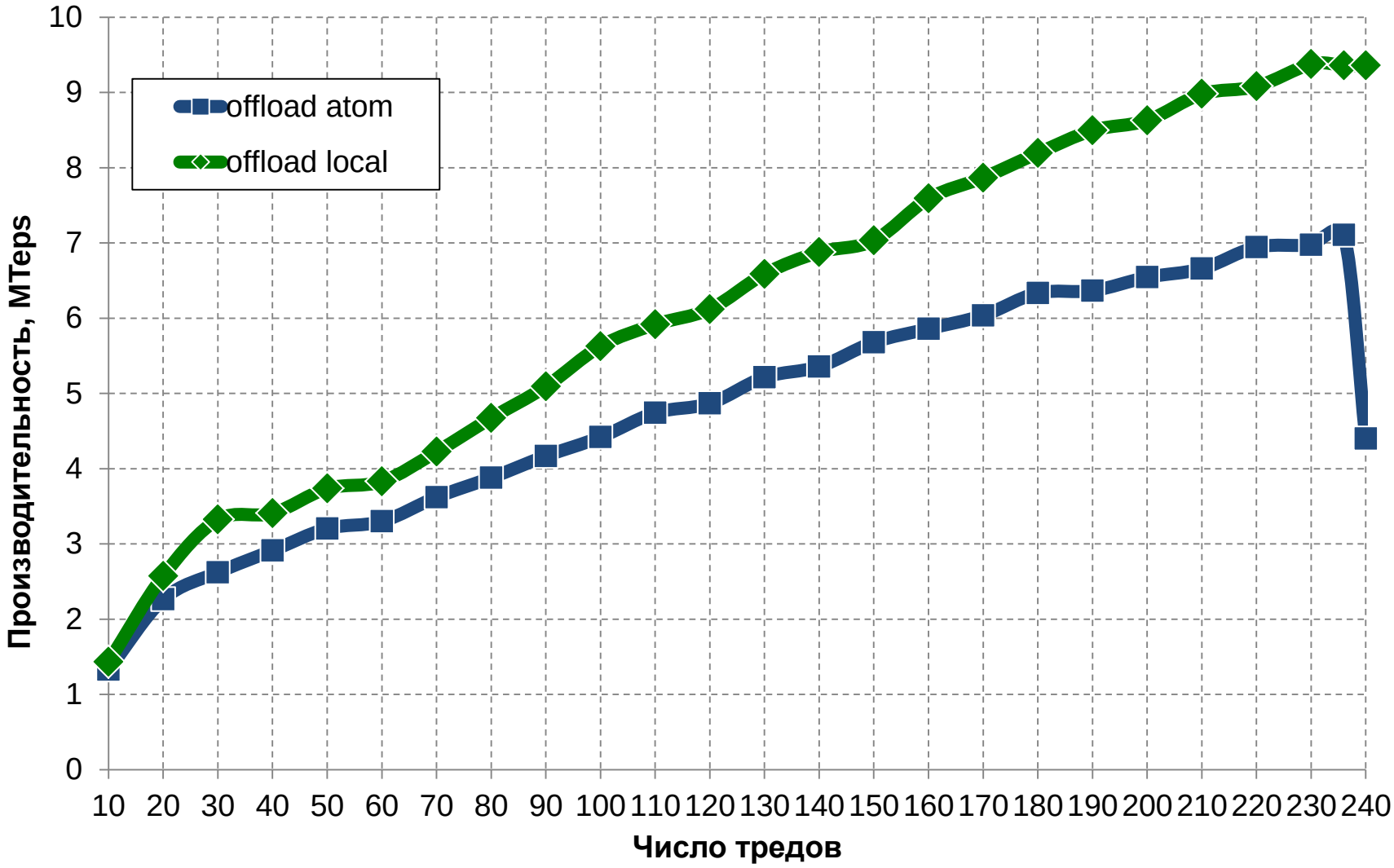
Алгоритм Противоход, версия local

$Q_{counter} = 1$
Q[0] = root
marked[root] = 1
while $Q_{counter} > 0$
    $Q_{next_counter} = 0$
    #pragma omp parallel for
    for all vertex ∈ Q **do**
        for all w: (vertex, w) ∈ E **do**
            if marked[w] == 0 **then**
                Q_{next}[**__sync_fetch_and_add**($Q_{next_counter}$, 1)] = w
                marked[w] = 1
                mark_correctCell(w)
            endif
        end for
    end for
    swap(Q, Q_{next}) // *обмен Q и Q_{next}*
end while
Fix_All_Cells // **корректировка всех помеченных нормалей**

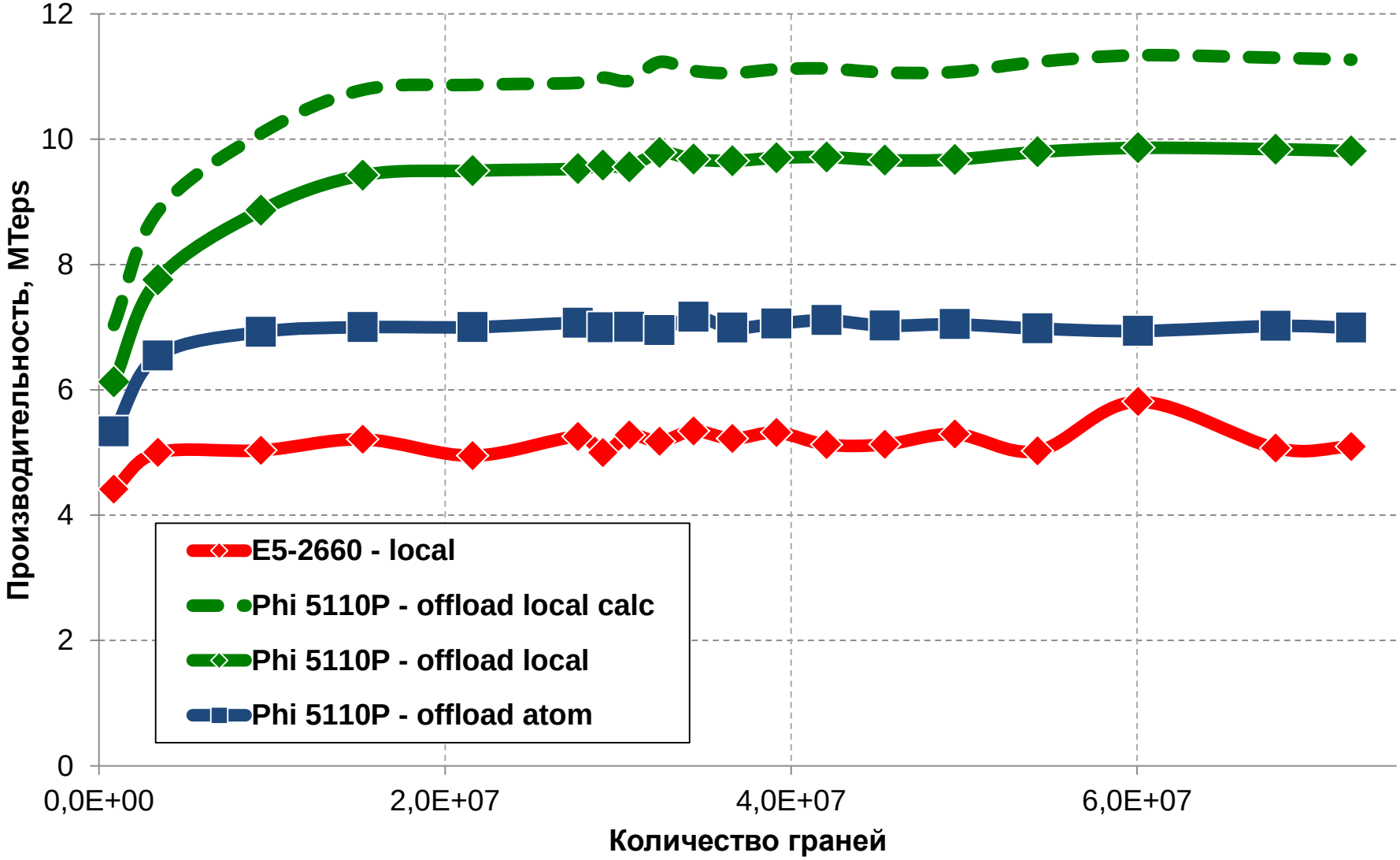


Массив Faces, элементы {0, 1, 2}

Производительность алгоритма Противоход, версия atom на Intel Xeon Phi 5110P (сетка - 3.5 ГБ)



Сравнение Intel Xeon E5-2660 с Intel Xeon Phi 5110P (сетки до 4.7 ГБ)



Выводы и планы развития

- Intel Xeon Phi 5110P опережает Intel Xeon E5-2660 на 41%
- Лучше избегать использовать атомарных операций

- Реализация с использованием MPI для обработки больших сеток

Спасибо за внимание!

Вопросы?