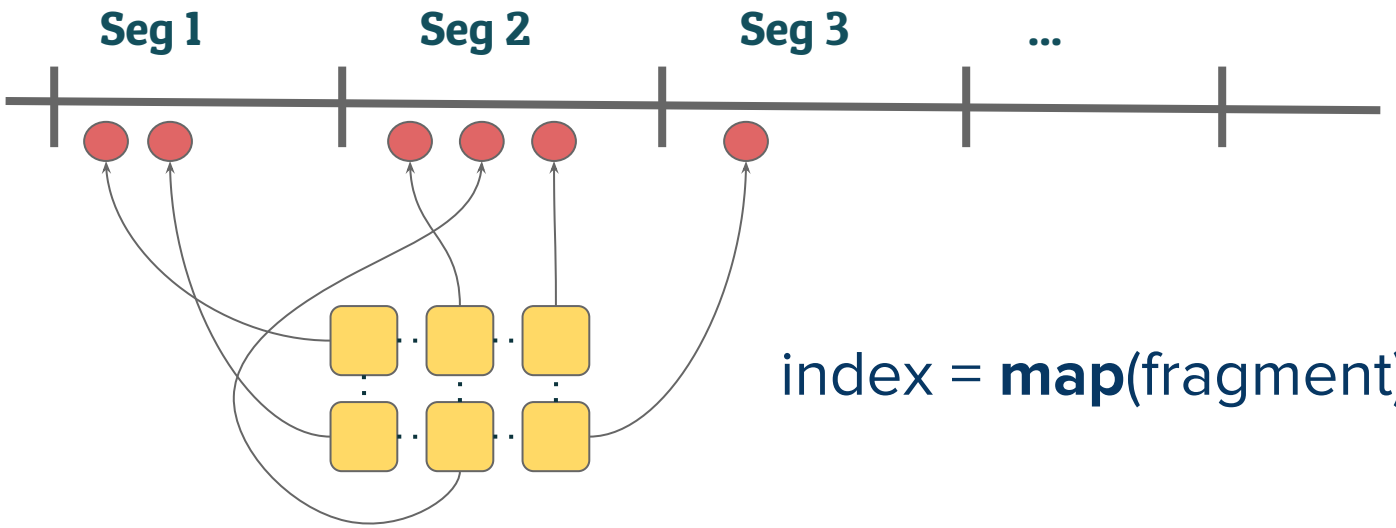
Распределенные алгоритмы с локальными взаимодействиями для управления данными в системе фрагментированного программирования LuNA

Щукин Г.А., ИВМиМГ СО РАН
Лаборатория Синтеза Параллельных Программ
Новосибирск 2017

Проблема

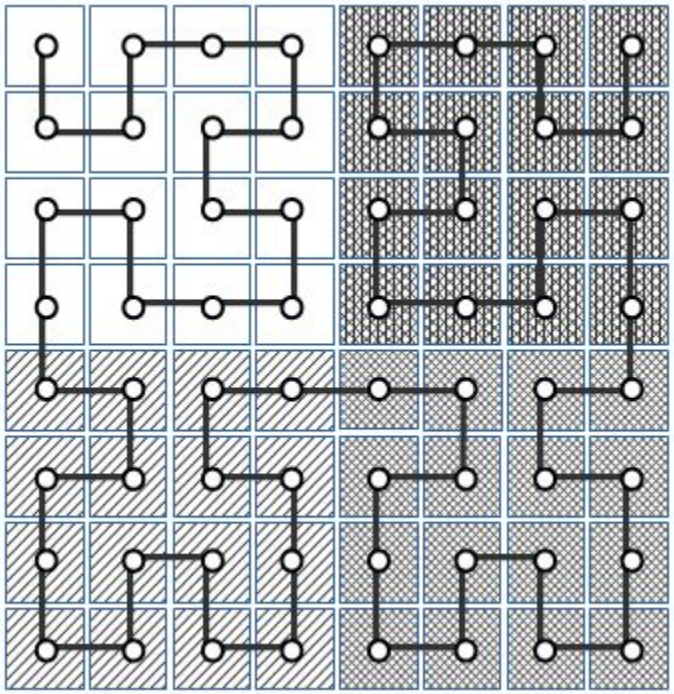
- Для параллельного программирования численных методов для крупномасштабных вычислительных систем с помощью системы LuNA требуются эффективные алгоритмы распределения данных и динамической балансировки нагрузки
- Алгоритмы должны быть распределенными и с локальными взаимодействиями (для масштабируемости)
- Алгоритмы должны учитывать топологию вычислительной сети и информационные зависимости в задаче

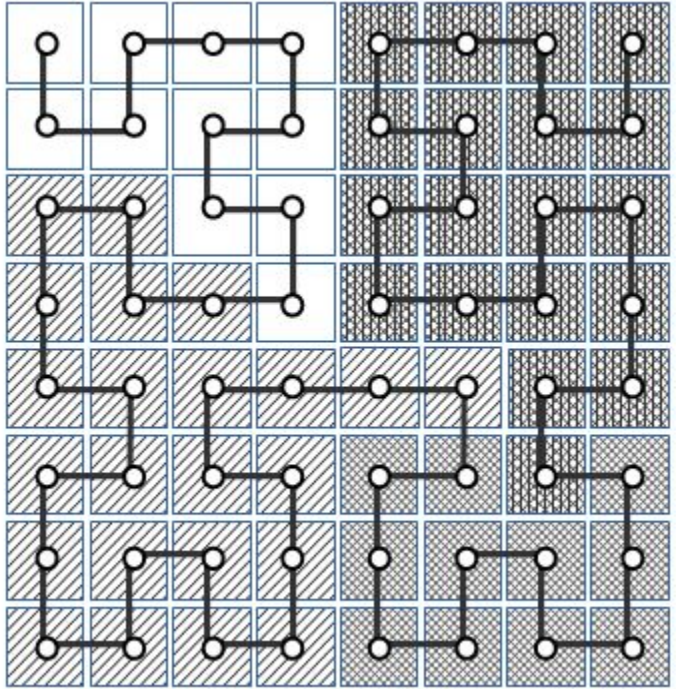
Алгоритм **Rope**



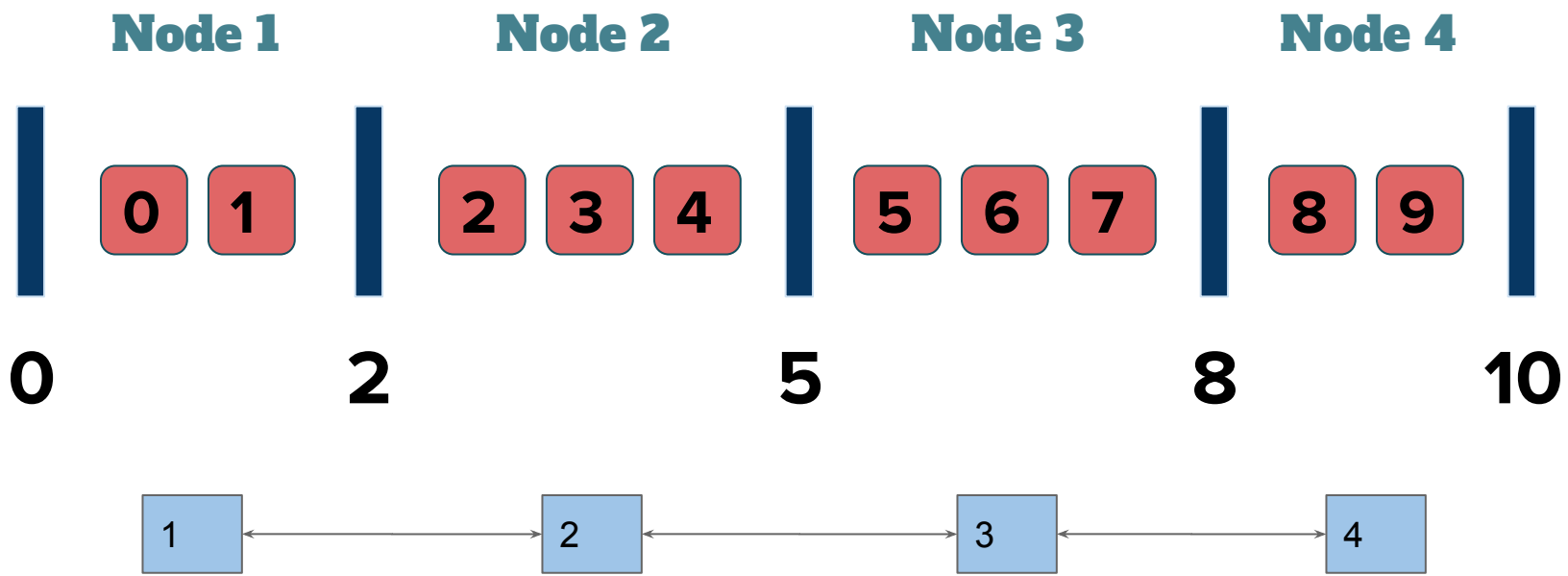
index = **map**(fragment)

Hilbert SFC

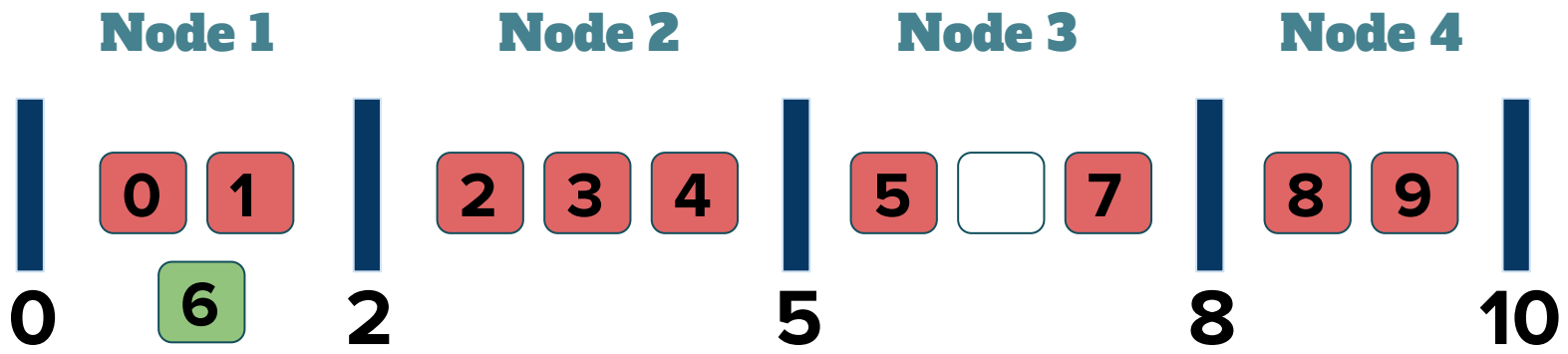




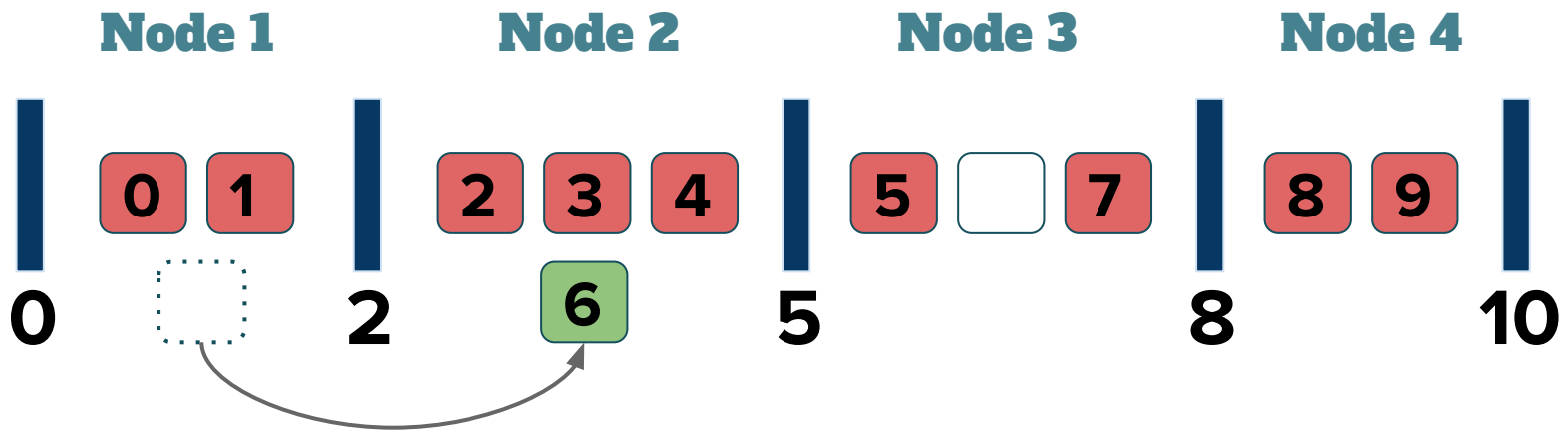
Распределение фрагментов данных



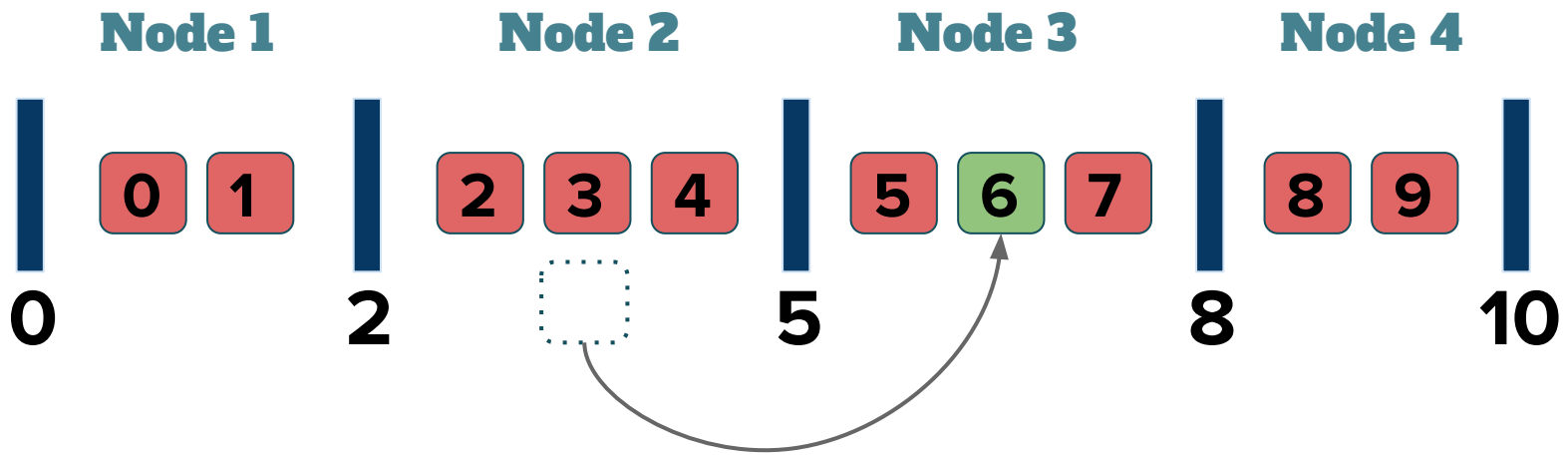
Распределение ФД (1)



Распределение ФД (2)



Распределение ФД (3)



Динамическая балансировка нагрузки

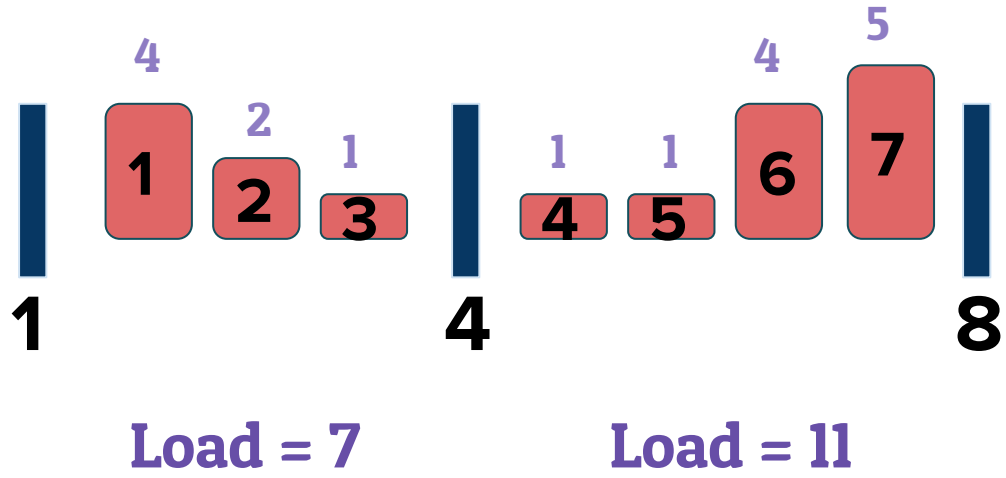
$$l_{coord} = \sum_{map(df)=coord} load(df)$$

$$L_{node} = \sum_{index \in node} l_{index}$$

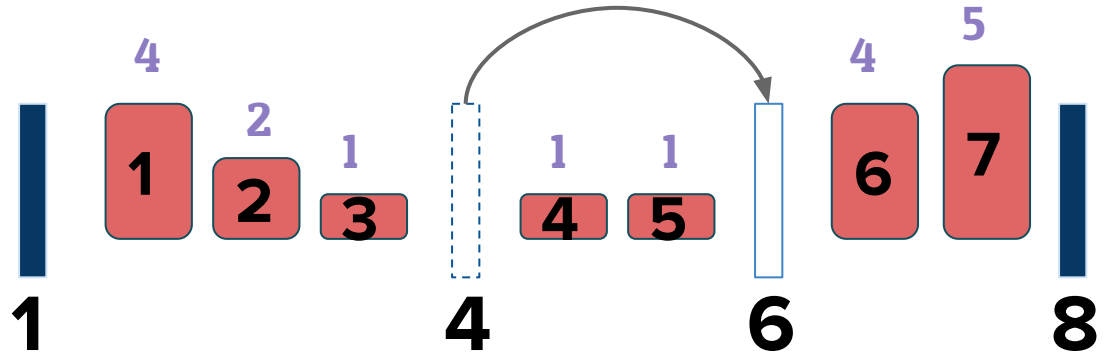
$$LAvg_{node} = \sum_{n \in nodes} \frac{L_n}{|nodes|},\ nodes = node \cup neigh(node)$$

$$overload_{node} = L_{node} - LAvg_{node}$$

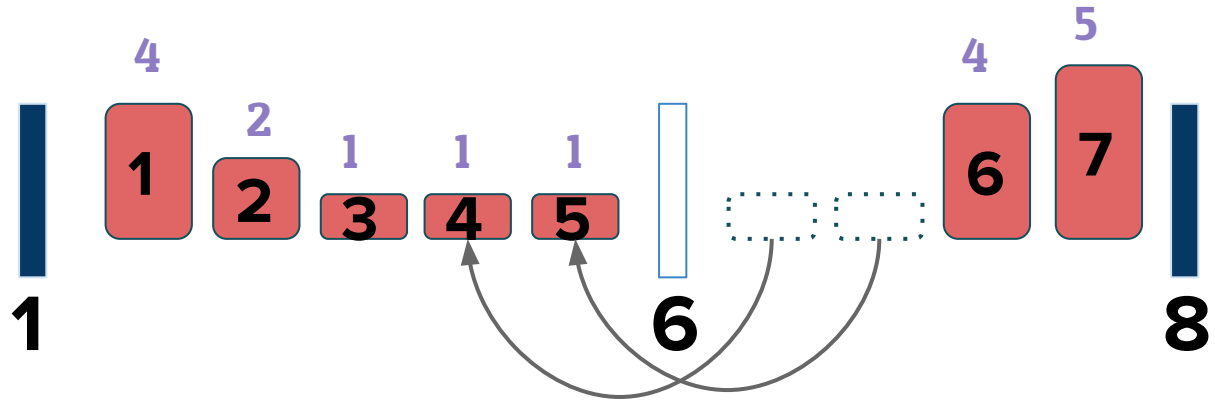
Балансировка нагрузки (1)



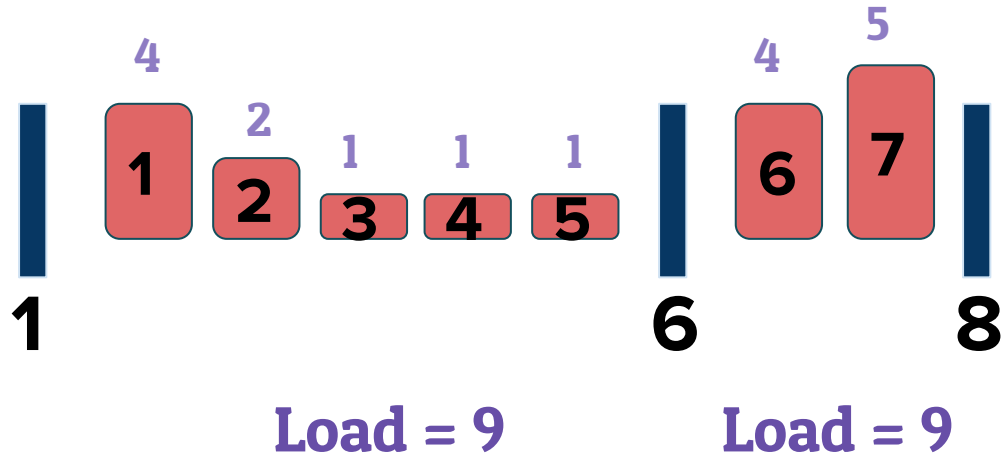
Балансировка нагрузки (2)



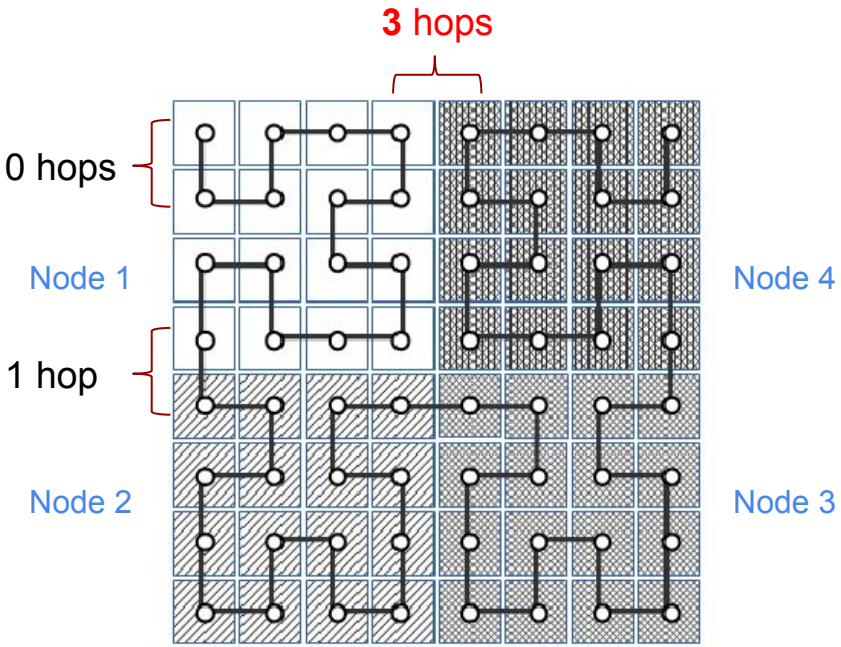
Балансировка нагрузки (3)



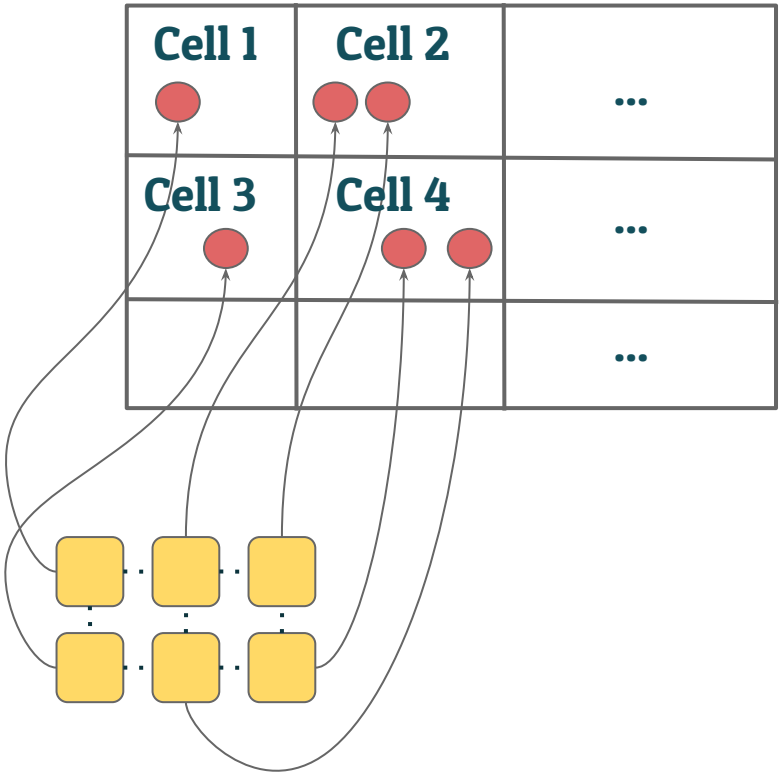
Балансировка нагрузки (4)



Недостатки одномерного отображения

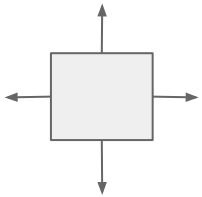


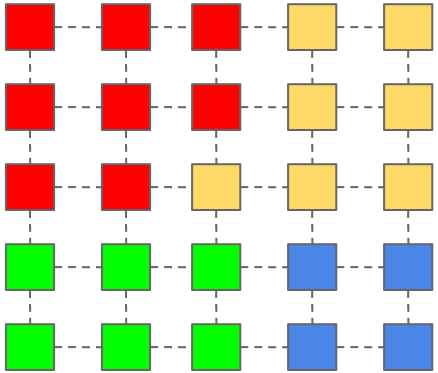
Алгоритм PATCH

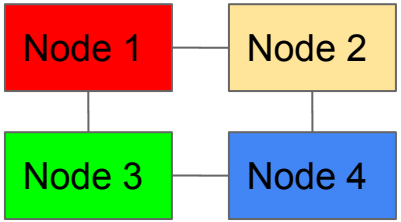


coord = **map**(fragment)

Многомерная сетка ячеек

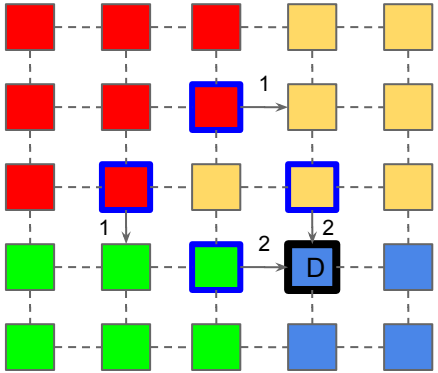


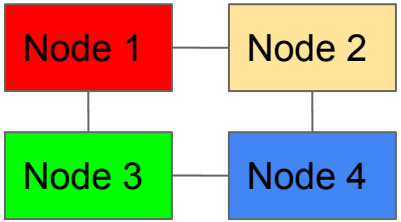




Решетка ПЭ

Распределение и поиск ФД





Динамическая балансировка нагрузки

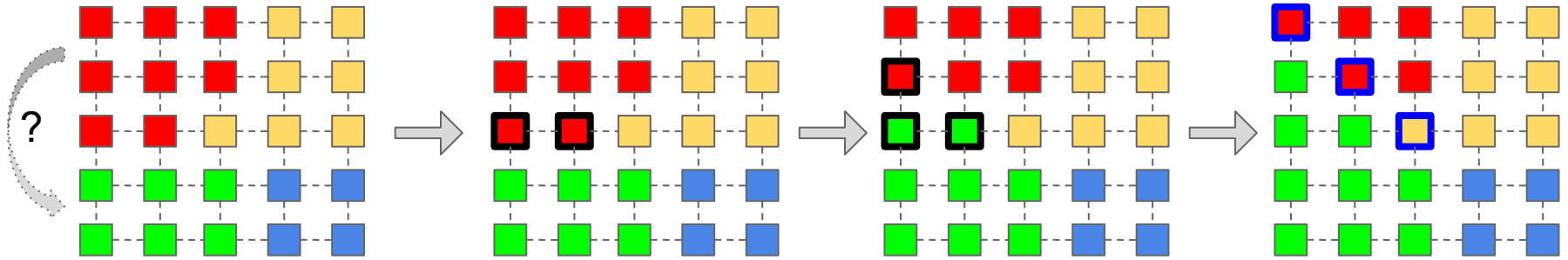
1. На каждом узле через заданный интервал времени выполнить шаг 2.
2. Если нагрузка узла больше средней (порог дисбаланса превышен), перейти на шаг 3, иначе ничего не делать.
3. Выбрать группу ячеек и передать ее на недогруженный соседний узел. Если еще остались недогруженные соседние узлы и текущий узел все еще перегружен, повторить шаг 3.

Алгоритм выбора ячеек для передачи

Задача: узлу Ns нужно отдать нагрузку L узлу Nd

1. На узле Ns выбрать начальную ячейку, смежную с Nd
2. Пока нагрузка выбранной группы ячеек меньше L, повторять шаг 2, иначе перейти на шаг 4
3. Добавить к группе такую новую смежную ячейку, чтобы число смежных ребер с узлом Nd было минимальным
4. Передать выбранную группу ячеек на узел Nd, обновить информацию о смежности для всех зависимых узлов

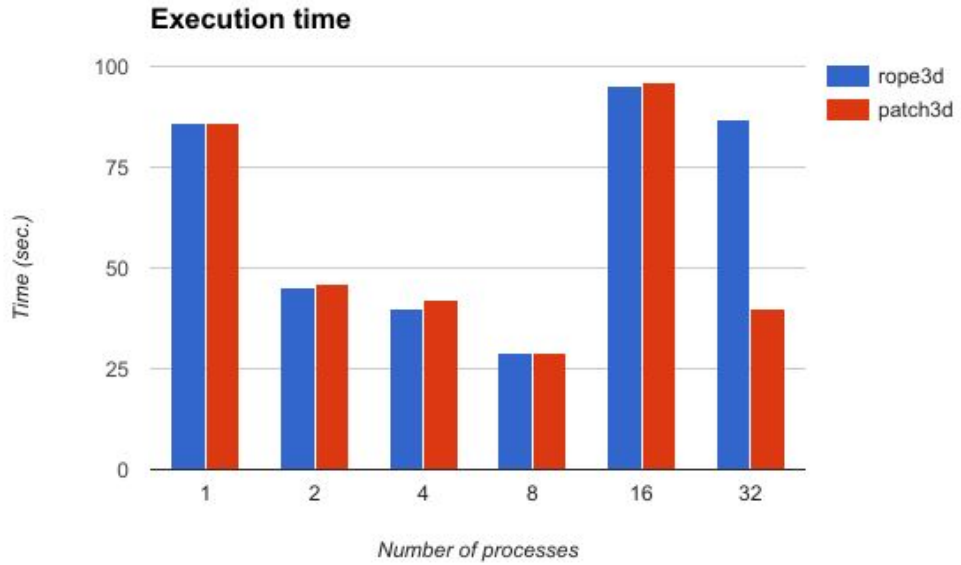
Передача ячеек: пример



Механизм транзакций

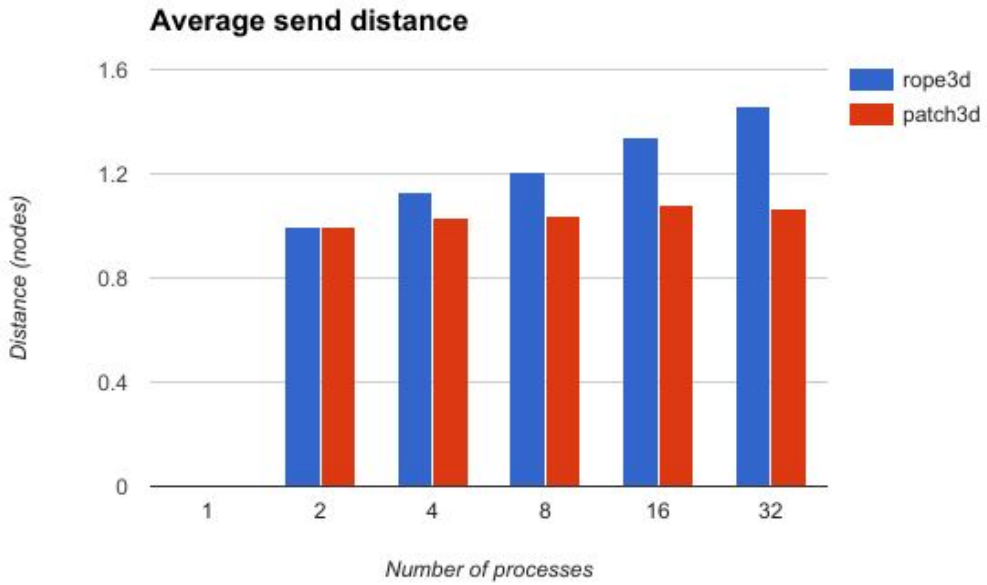
- Транзакция - передача ячеек от одного узла другому
- Каждый узел может быть как отправителем, так и получателем
- Необходимо синхронизовать получение и отправку:
  - Активные запросы
  - Использование приоритета транзакций

Тесты

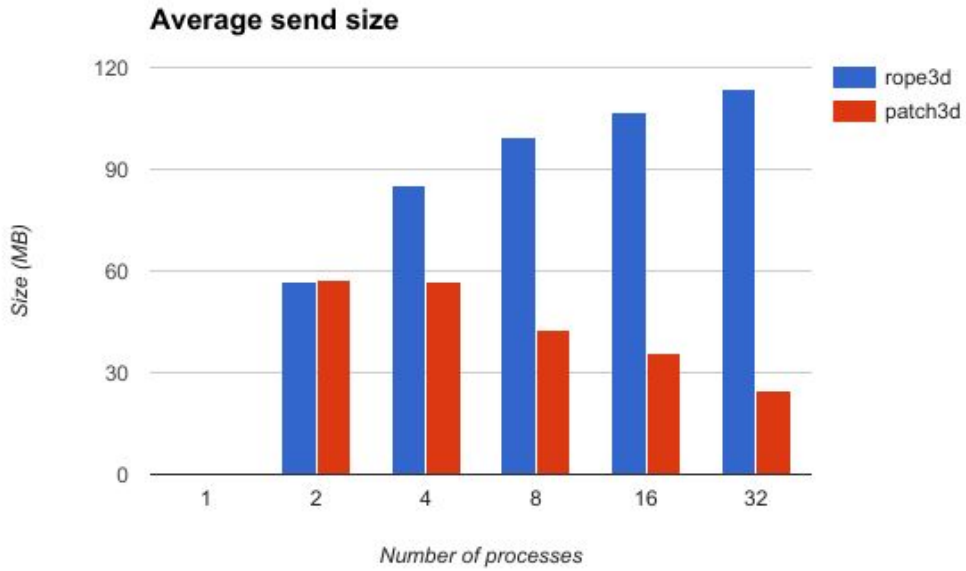


Пуассон,
сетка 256*256*256,
фрагментация 8*8*8

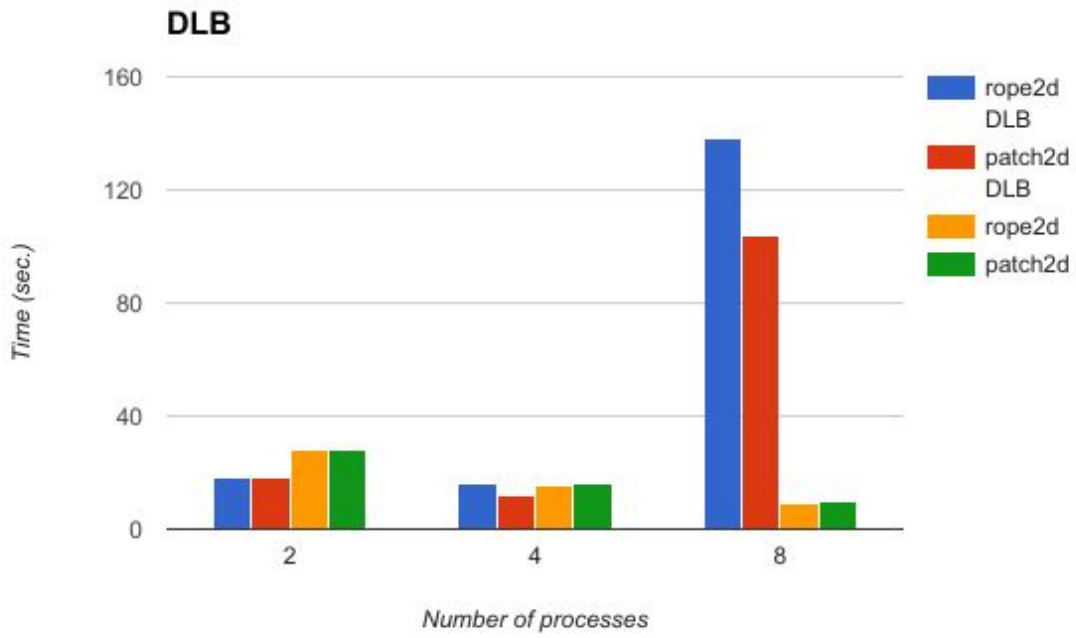
Тесты


Average send distance
1.6
1.2
0.8
0.4
0
Distance (nodes)
1
2
4
8
16
32
Number of processes
rope3d
patch3d

Тесты


Average send size
120
90
60
30
0
Size (MB)
1
2
4
8
16
32
Number of processes
rope3d
patch3d

Тесты


DLB
160
120
80
40
0
Time (sec.)
2
4
8
Number of processes
rope2d DLB
patch2d DLB
rope2d
patch2d

Выводы

- Разработаны распределенные алгоритмы с локальными коммуникациями для распределения данных и динамической балансировки нагрузки в системе LuNA
- Проведено тестирование алгоритмов
- Планируется дальнейшая доработка алгоритмов, в особенности динамической балансировки нагрузки

Спасибо!