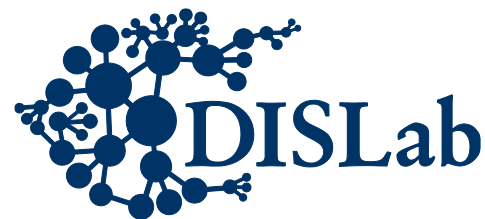


Параллельная обработка больших графов

Занятие 7

А.С. Семенов

dislab.org



Виды графов

Виды графов. Случайные графы

- Random, Random Uniform, Erdos Renyi (1959)

• Обозначение: $G(N, M)$, $G(N, p)$

N – количество вершин,

M – количество ребер

p – вероятность ребра между любыми вершинами

Свойства:

- Пусть $p = (c \ln N)/N$

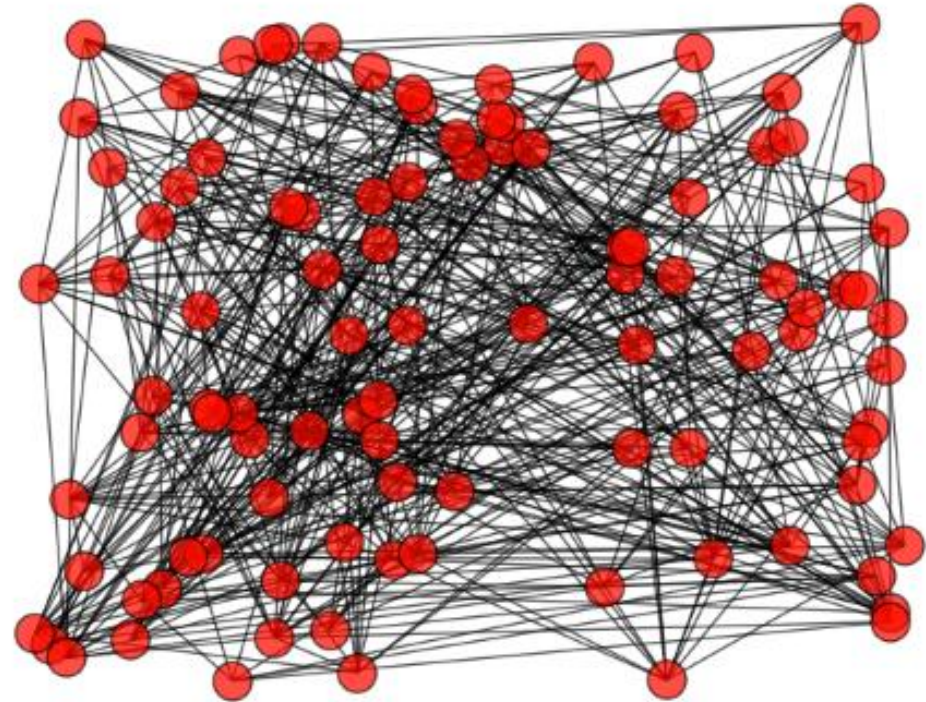
- если $c > 1$, то почти всегда случайный граф связан.

- если $c < 1$, то почти всегда случайный граф не связный.

- Пусть $p = c / N$

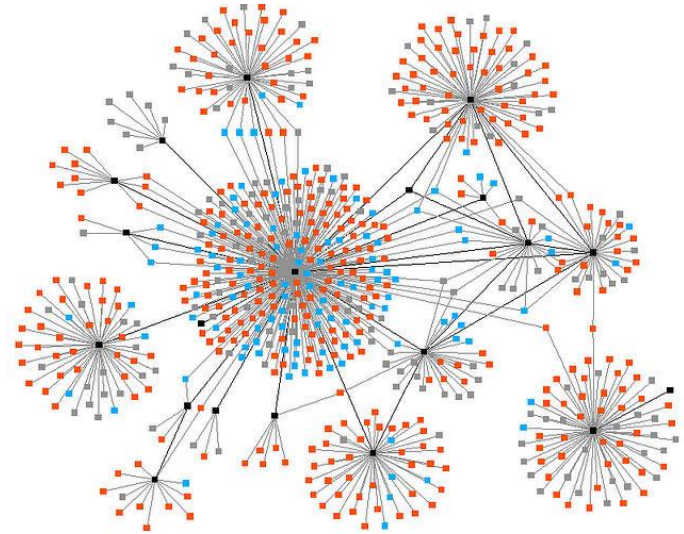
- при любом $c < 1$ существует такая константа $\beta = \beta(c)$, что почти наверное каждая компонента связности случайного графа имеет не более $\beta \ln n$ вершин.

- при любом $c > 1$ существует такая константа $\gamma = \gamma(c) \in (0, 1)$, что почти наверное среди компонент случайного графа есть одна (гигантская), число вершин которой не меньше γn .

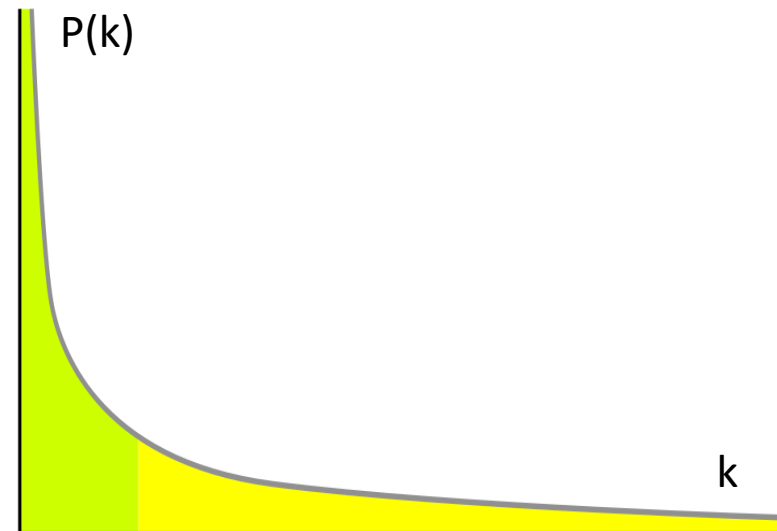


Виды графов. Степенной закон

- WWW, Социальные сети, Биоинформатика
- Графы small-world – графы, для которых
 $L \sim \log N$

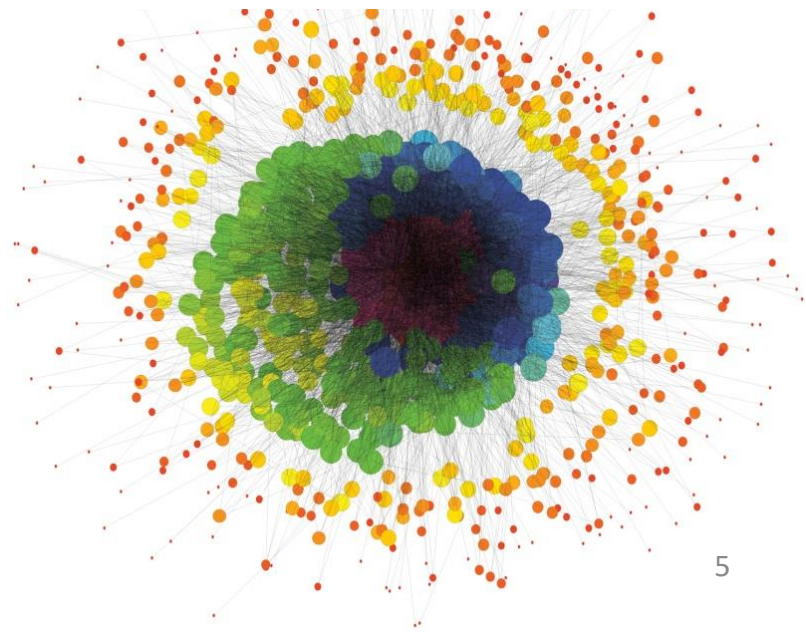
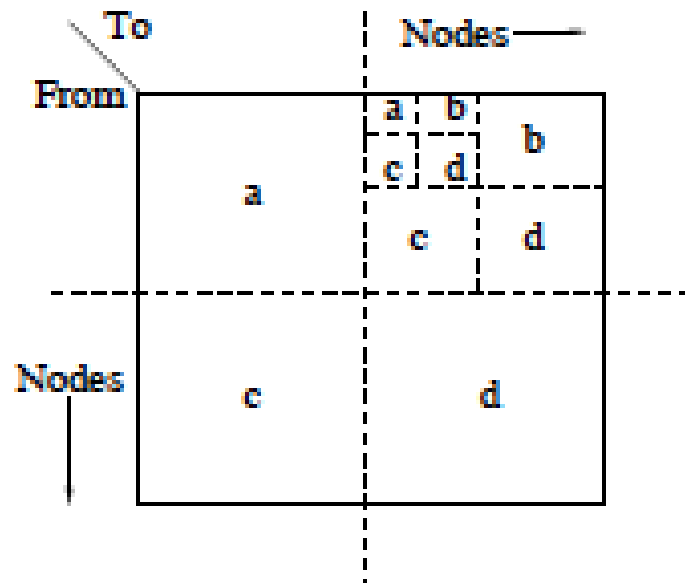


- scale-free – графы,
доля $P(k) \sim k^{-\tau}$, $2 < \tau < 3$
 k – связность вершины
 $L \sim \log \log N$



Виды графов. RМAТ-граф

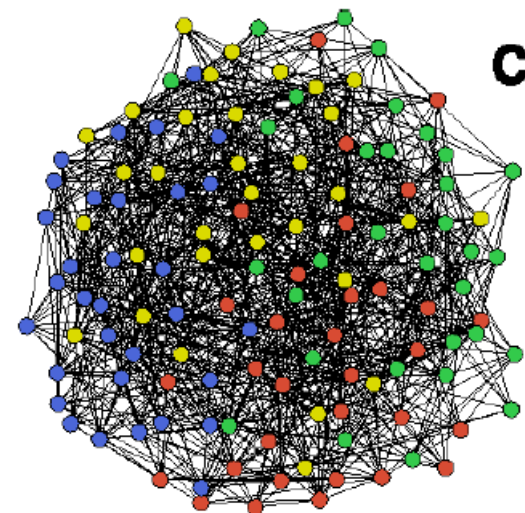
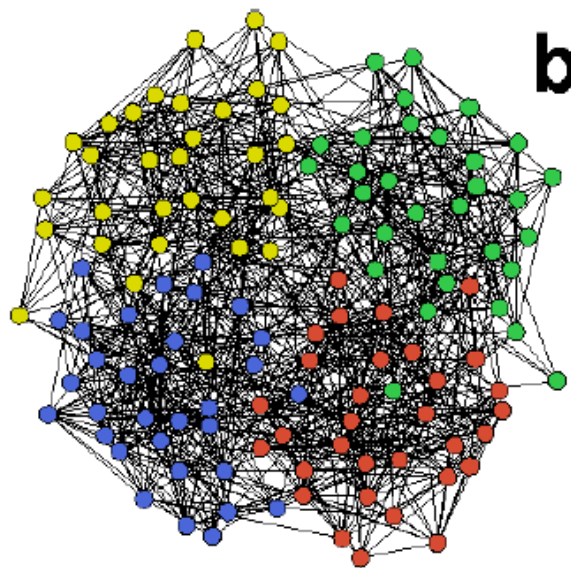
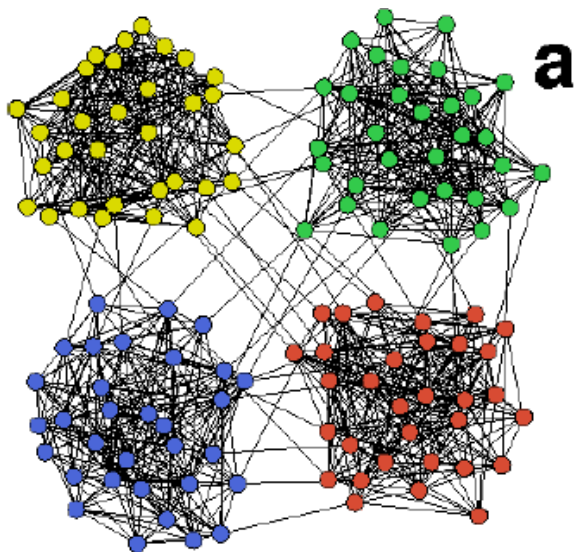
- $a+b+c+d = 1$
- Сообщества:
 - a и d – сообщества
 - b и c – связи между ними
 - наличие «подсообществ»
- может быть scale-free при $a \geq d$
- случайная перестановка вершин



Виды графов. LFR*-граф

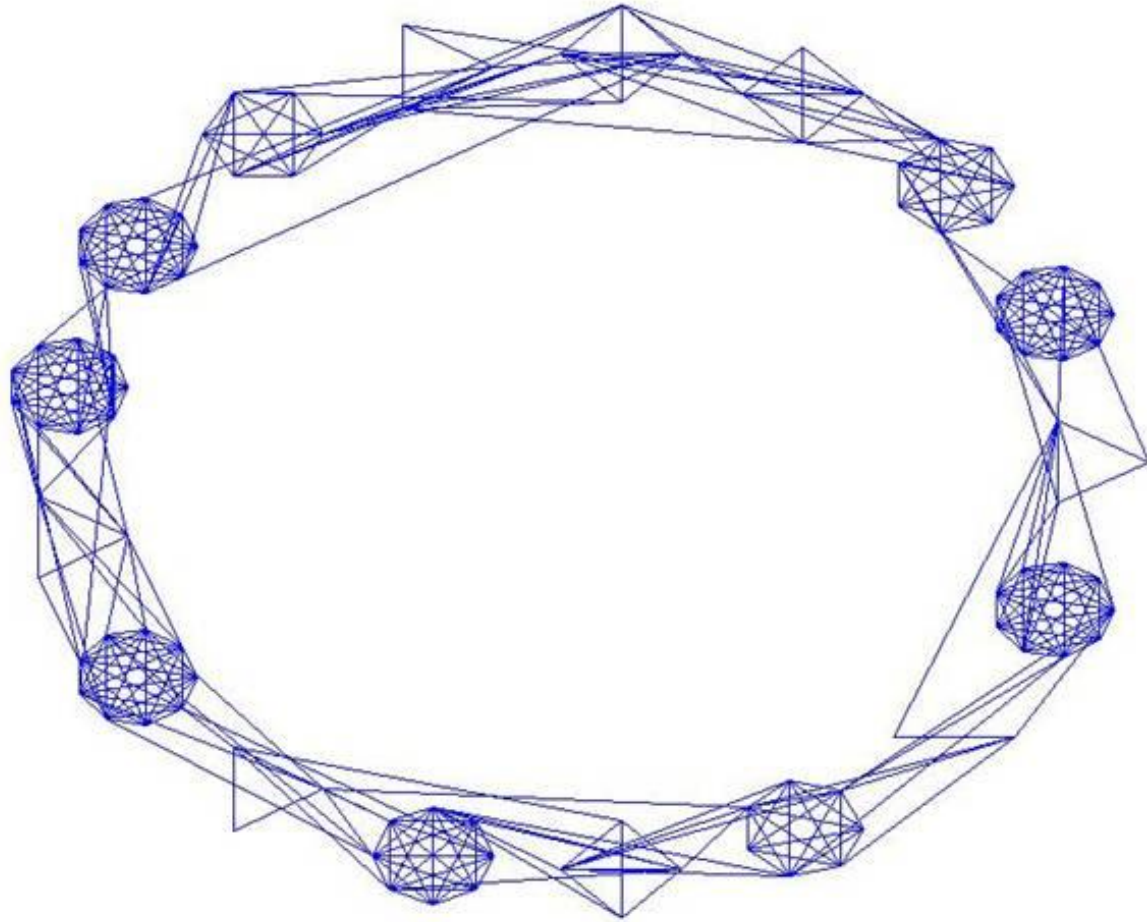
- **Параметры:**

- $m_i \in [0;1]$, показывает количество связей вне сообщества
- com_tau – показатель степени в законе распределения размеров сообществ
- deg_tau – показатель степени в законе распределения степеней вершин



Виды графов. SSCA2-граф

- Равномерное распределение случайных параметров
- случайная перестановка вершин

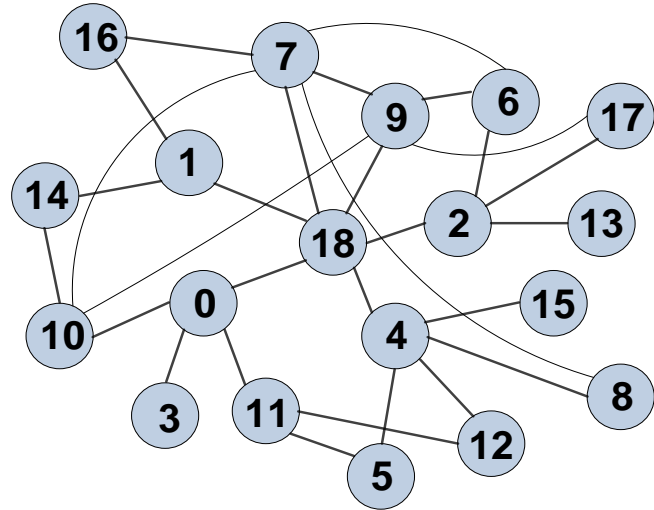


Проблемы и подходы к решению задач обработки графов в рамках одного вычислительного узла

Проблемы анализа больших графов

- **Data-driven computations.** Зависимость вычислений от данных (топологии графа). Невозможность применения методов статического распараллеливания вычислений.
- **Unstructured problems.** Работа с нерегулярными, неструктурированными данными, трудность распараллеливания.
- **Poor locality.** Низкая пространственно-временная локализация обращений к памяти.
- **High data access to computation ratio.** Преобладание команд доступа к памяти над командами выполнения арифметических операций.

Представление графа



	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
0				1							1	1							1
1															1		1		1
2							1							1				1	1
3	1																		
4						1		1				1			1				1
5					1							1							
6			1					1		1									
7							1		1	1	1							1	1
8					1			1											
9							1	1			1							1	1
10	1							1		1					1				
11	1					1							1						
12					1							1							
13			1																
14		1									1								
15					1														
16		1						1											
17			1							1									
18	1	1	1		1			1		1									

Форматы представления разреженных матриц

- Доля ненулевых элементов мала

Можно хранить только позиции и значения ненулевых элементов

- **Compressed Row Storage (CRS)**
- **Coordinate list (COO)**
- **DIA**
- **ELLPACK**
- **JDS**
- **SELLPACK**
- **Оптимизированный под задачу**

Внутреннее представление Compressed Row Storage (CRS)

Sparse Matrix

10	0	0	0	-2
3	9	0	0	0
0	7	8	7	0
3	0	8	7	5
0	8	0	9	13

Row pointer array

0	2	4	7	11	14
---	---	---	---	----	----

rowsIndices

Column indices array

0	4	0	1	1	2	3	0	2	3	4	1	3	4
---	---	---	---	---	---	---	---	---	---	---	---	---	---

endV

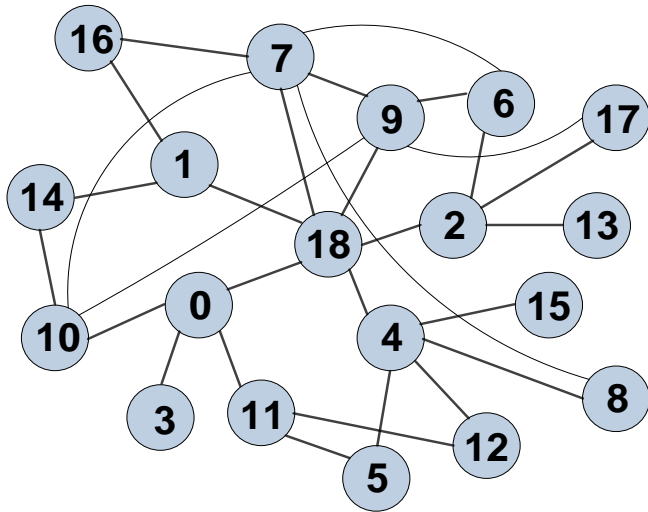
Values array

10	-2	3	9	7	8	7	3	8	7	5	8	9	13
----	----	---	---	---	---	---	---	---	---	---	---	---	----

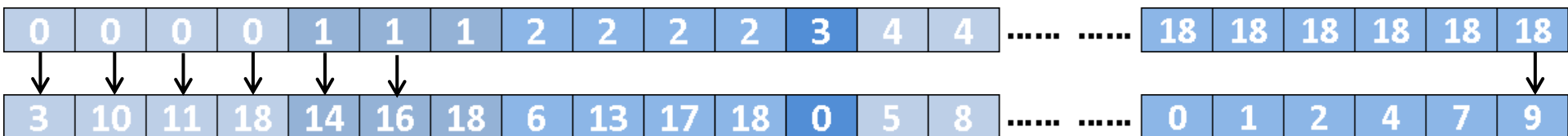
weights

```
for (int u = 0; u < G->n; u++) {  
    for (int j = G->rowsIndices[u]; j < rowsIndices[u+1]; j++) {  
        const int v = G->endV[j];  
        const int w = G->weights[j];  
        // обработка ребра u->v  
    }  
}
```

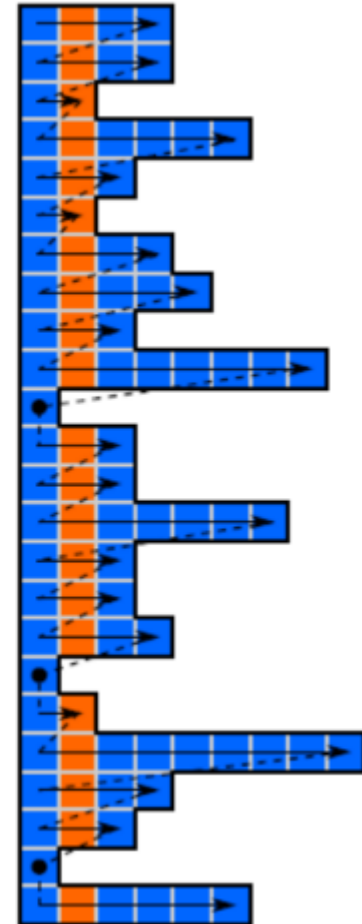
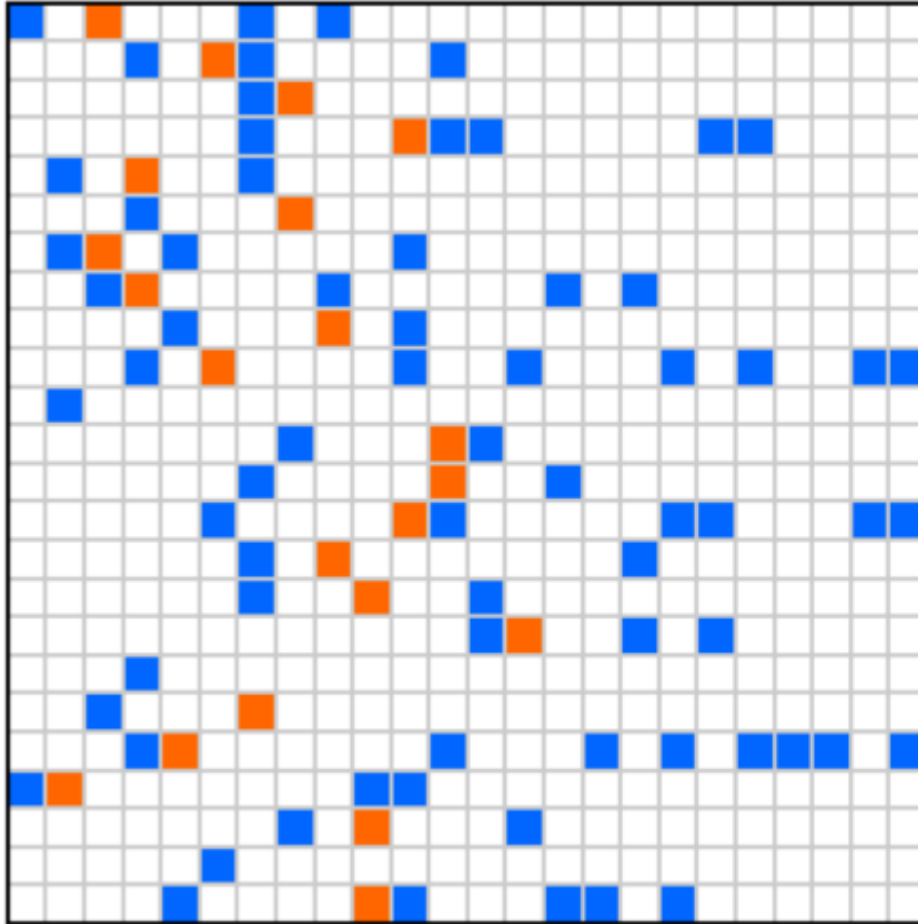
Coordinate list (COO)



	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
0				1							1	1							1
1															1		1		1
2							1							1				1	1
3	1																		
4						1		1					1			1			1
5					1							1							
6			1					1		1	1								
7							1		1	1	1						1		1
8					1			1											
9							1	1			1							1	1
10	1							1		1					1				
11	1					1								1					
12					1							1							
13			1																
14	1										1								
15					1														
16		1							1										
17			1								1								
18	1	1	1	1	1			1	1										



CRS



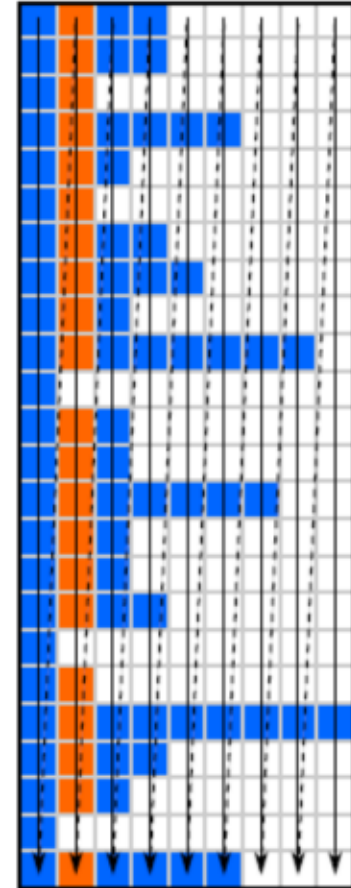
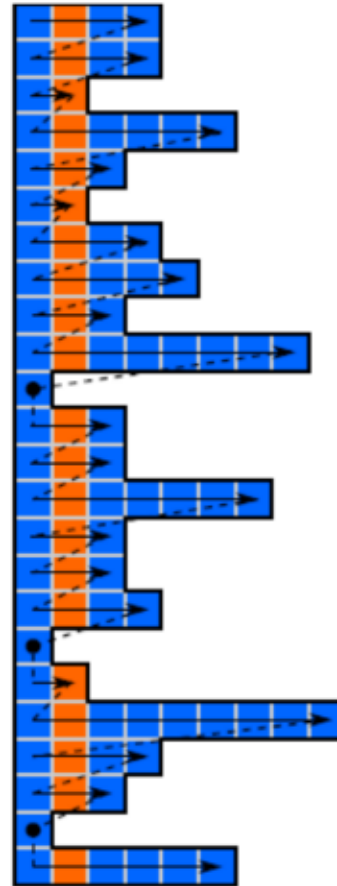
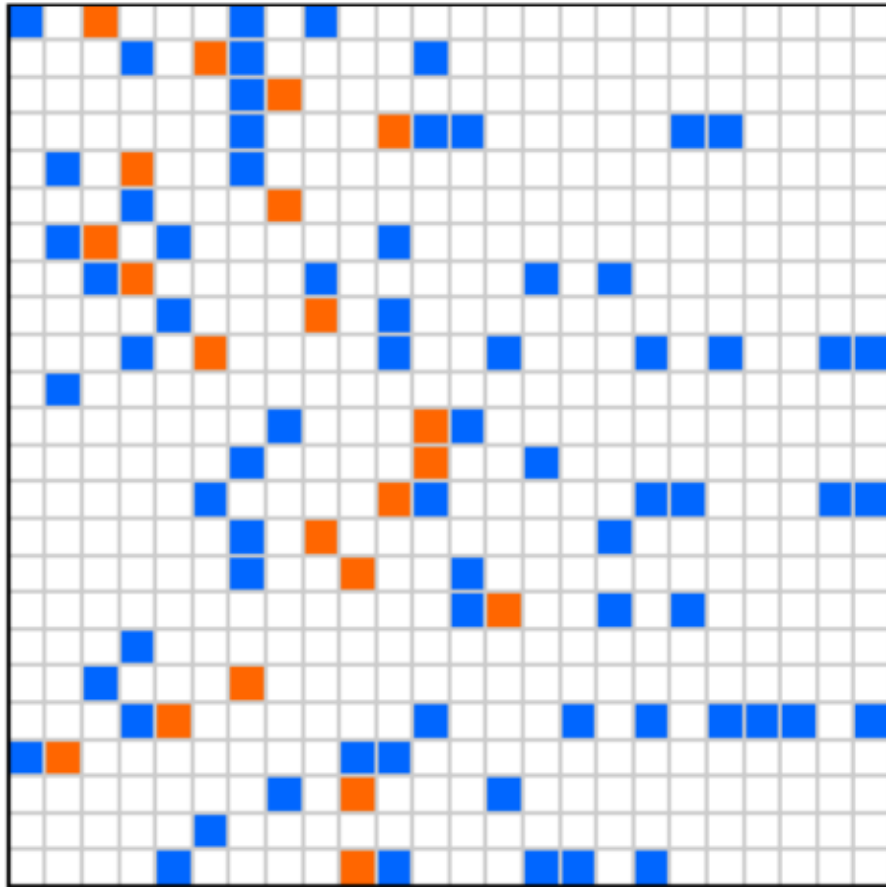
CRS – SIMD

```
1 for(i = 0; i < N; ++i) {
2   for(j = rpt[i]; j < rpt[i+1]; ++j) {
3     y[i] += val[j] * x[col[j]];
4   }
5 }
```

```
1 for(i = 0; i < N; ++i)
2 {
3   tmp0 = tmp1 = tmp2 = tmp3 = 0.;
4   for(j = rpt[i]; j < rpt[i+1]; j+=4)
5   {
6     tmp0 += val[j+0] * x[col[j+0]];
7     tmp1 += val[j+1] * x[col[j+1]];
8     tmp2 += val[j+2] * x[col[j+2]];
9     tmp3 += val[j+3] * x[col[j+3]];
10  }
11  y[i] += tmp0+tmp1+tmp2+tmp3;
12  // remainder loop
13  for(j = j-4; j < rpt[i+1]; j++)
14    y[i] += val[j] * x[col[j]];
15 }
```

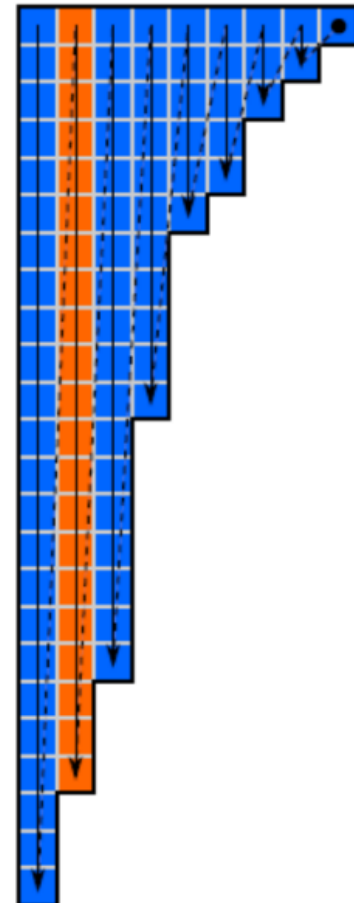
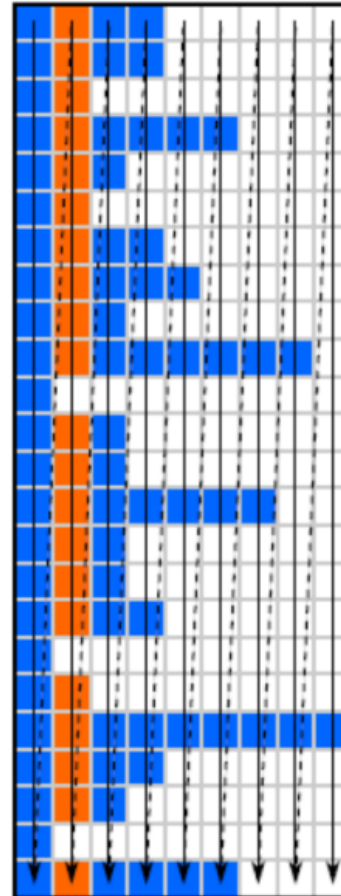
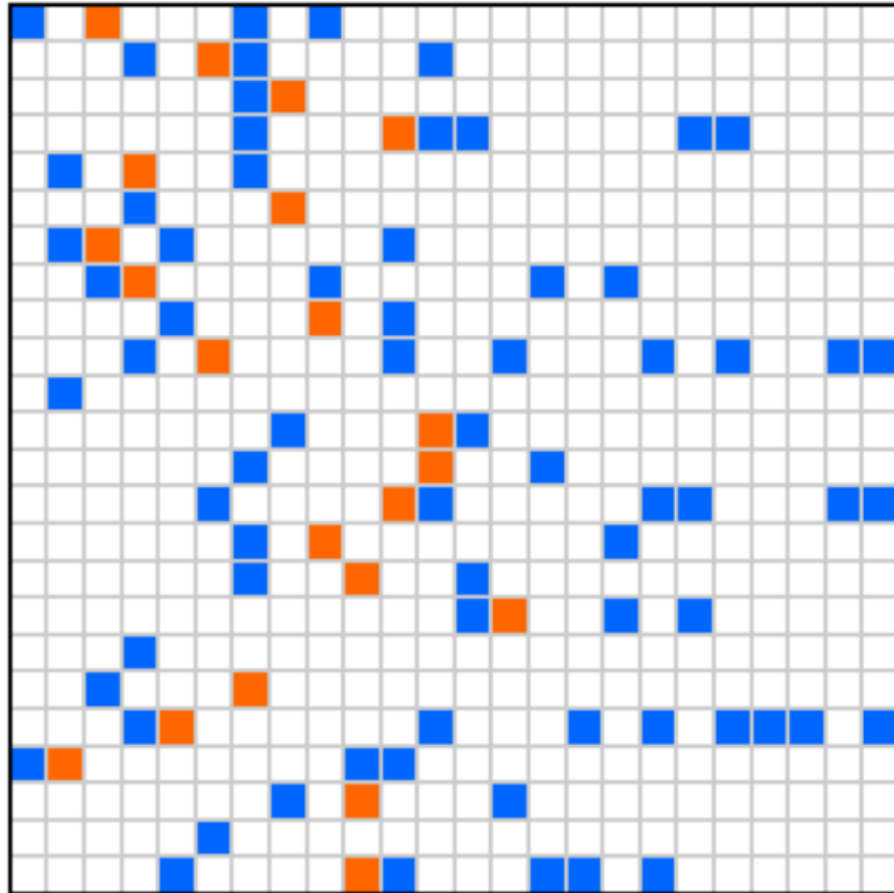
- Плохо, если строки короткие, тогда доля редукции (line 11) и остатка велика

ELLPACK



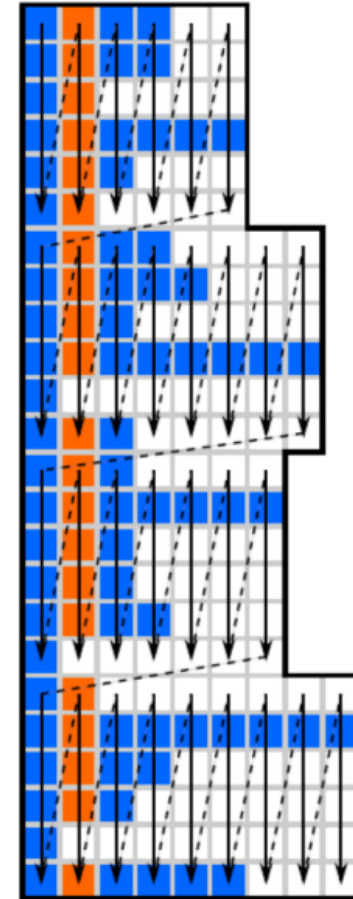
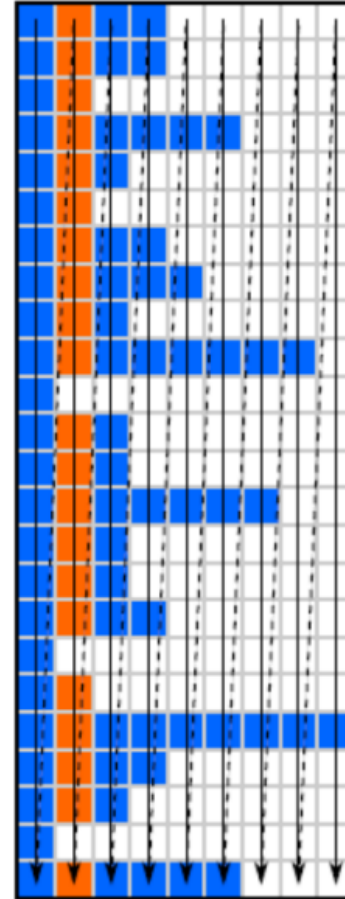
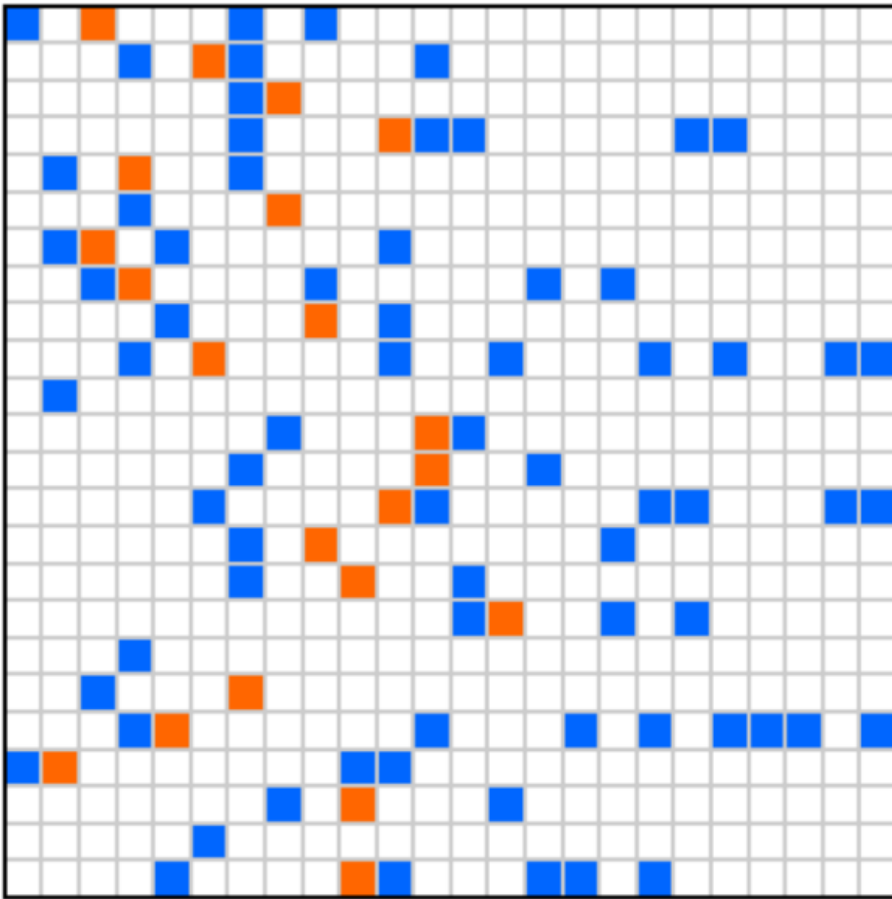
- Элементы хранятся в плотном массиве по столбцам (длинным), хорошо для попадания в кэш
- Нули хранятся явно (zero padding)
- Хорошо, если все строки примерно одинаковой длины, наоборот - плохо

JDS – Jagged Diagonal Format



- Сортировка по длинам строк
- Плохо то, что доступ в x ($y = Ax$) может быть нерегулярным, плохое использование кэша

Sliced ELLPACK (SELLPACK)



- Хранит не так много 0 по сравнению с ELLPACK
- Убирает проблему нерегулярного доступа к x ($y = Ax$)

Общие методы программной оптимизации для SMP x86 систем

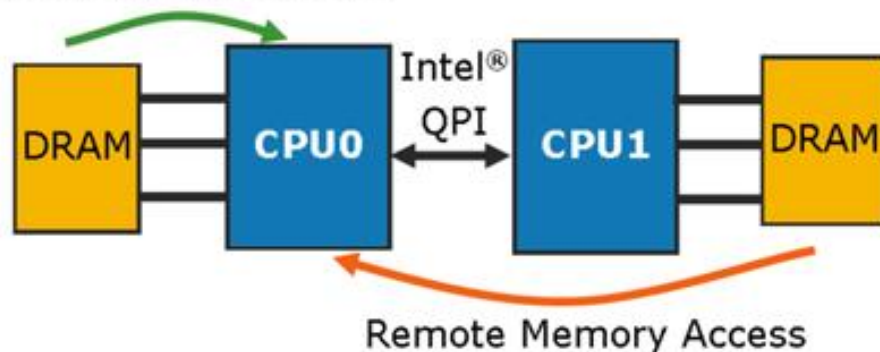
Архитектура вычислительного узла

- 2 сокета с Intel Xeon E5-2683 v3 2.00 GHz
- Turbo до 3 GHz
- кэш 35 МБ
- NUMA
- 14 ядер на сокет
- Hyper-Threading (в сумме 56 потоков)
- QPI 9.6 GT/s
- 64 GB

- Intel Xeon Phi 5110P
- Векторизация, привязка потоков, локализация данных

SMP узел:

Local Memory Access



Задержка доступа к памяти, тактов

	NODE
Кэш L1	4
Кэш L2	15
Кэш L3	40
DRAM	170
NUMA	400