

НИЖЕГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИМ. Н.И. ЛОБАЧЕВСКОГО
ИНСТИТУТ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ, МАТЕМАТИКИ И МЕХАНИКИ





Нижегородский государственный университет им. Н.И. Лобачевского
Институт информационных технологий, математики и механики

ПАРАЛЛЕЛЬНАЯ ОБРАБОТКА ГРАФОВ

Лекция 5. Алгоритмы разделения графов

Пирова А.Ю.
Кафедра ВВиСП

Содержание

- ❑ Предметные области
- ❑ Постановка задачи
- ❑ Методы разделения графа
- ❑ Многоуровневый метод разделения графа
- ❑ Бенчмарки
- ❑ Библиотеки разделения графов
- ❑ Результаты экспериментов
- ❑ Заключение

Постановка задачи

- Пусть дан ненаправленный граф $G = (V, E, W_e, W_v)$ с неотрицательными весами ребер $W_e: E \rightarrow \mathbf{R}_{\geq 0}$, весами вершин $W_v: V \rightarrow \mathbf{R}_{\geq 0}$, и число $k \in \mathbf{N}_{\geq 1}$.
- Задача разделения графа: найти разделение множества вершин $\Pi = (V_1, V_2, \dots, V_k)$ на такие непересекающиеся подмножества, что

$$\begin{cases} V_i \cap V_j = \emptyset, \forall i \neq j \\ \bigcup_{i=1}^k V_i = V \end{cases}$$

- Разделение должно быть *сбалансированным*:

$$|V_i| \leq (1 + \varepsilon) \left\lfloor \frac{|V|}{k} \right\rfloor, \forall i \in \{1, 2, \dots, k\}, \varepsilon \in \mathbf{R}_{\geq 0}$$

– $|V|$ - число вершин в V или суммарный вес вершин из V

Постановка задачи

- ❑ Вершина, смежная хотя бы с одной вершиной из другой части, называется *граничной*, а ребро, их соединяющее, *ребром разреза*.

$$u \in V_i, v \in V_j, (u, v) \in E, i \neq j$$

- ❑ Цель задачи разделения графа – найти разделение, для которого минимальна некоторая функция, оценивающая качество разделения.
- ❑ В большинстве приложений *качество разделения* оценивается по весу ребер разреза: чем меньше суммарный вес ребер разреза, тем разделение лучше

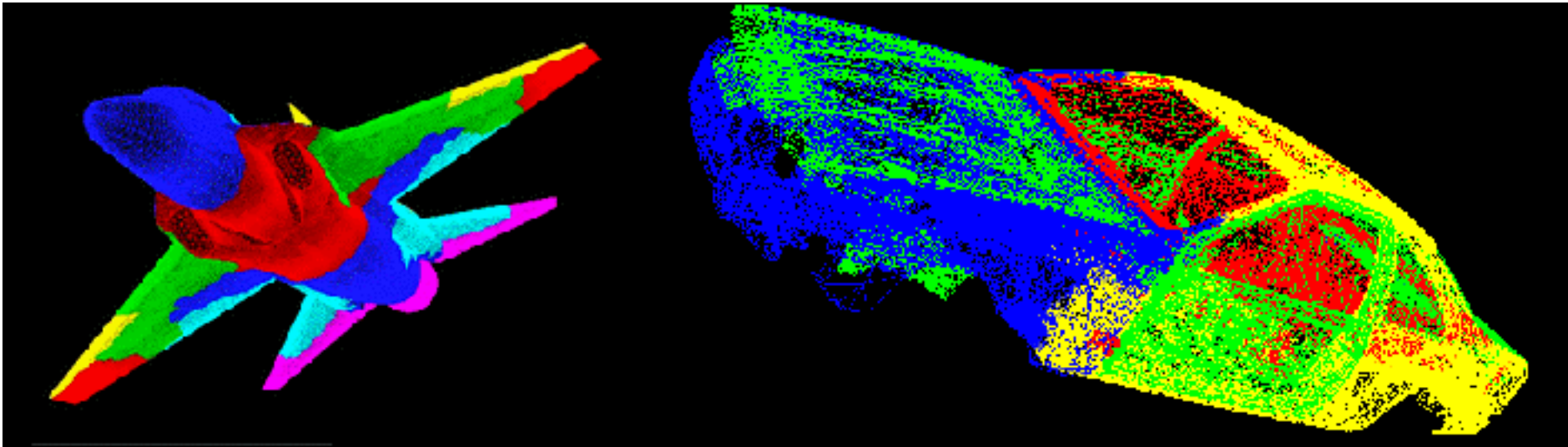
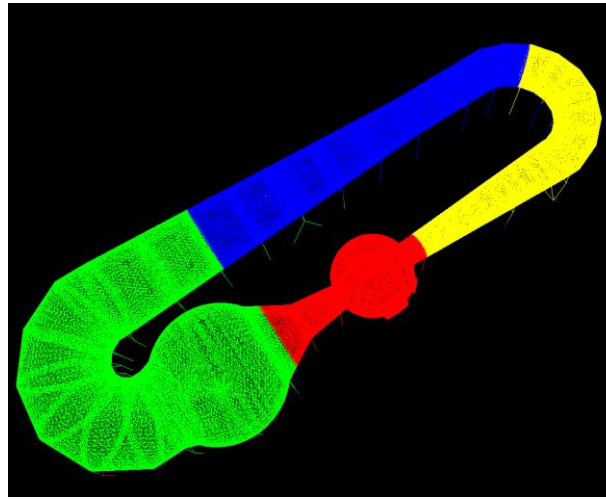
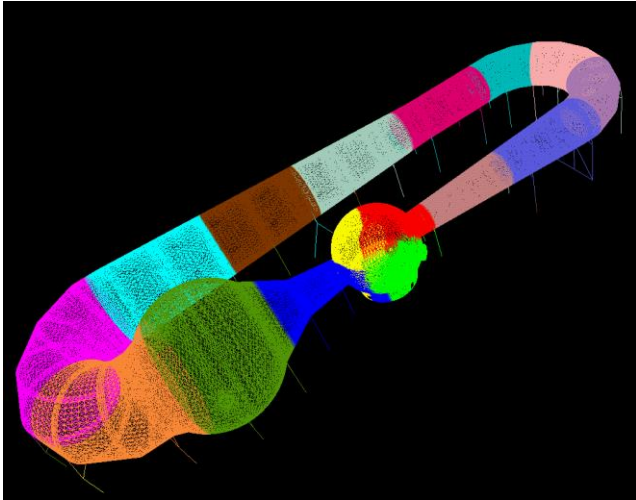
$$f: \sum_{(u,v) \in E_B} w(u, v) \rightarrow \min$$

❑ Задача NP-трудная

Предметные области

- ❑ Параллельные вычисления в моделировании
 - распределение данных по процессорам вычислительной системы
 - разделение вычислительных сеток
 - разделение графа для параллельной обработки
- ❑ Разделение сетей различной структуры
 - Транспортные сети
 - Электросети
 - Биологические системы
- ❑ Обработка изображений
- ❑ Разработка интегральных схем (VLSI design)

Примеры



К-разделение и бисекция

- К-разделение можно получить из 2-разделения методом *рекурсивной бисекции*: граф делится на 2 части, затем, рекурсивно, первая часть делится на $\lfloor k/2 \rfloor$ частей, вторая – на $\lfloor k/2 \rfloor$ частей и т.д. Всего потребуется выполнить $k - 1$ бисекцию.

Методы разделения графа

- ❑ Различные алгоритмы разделения графов разрабатываются с 1970х годов. Основные алгоритмы были разработаны в 1990х годах.
 - Растущее разделение (graph growing partitioning) – Джордж и Лю (1981), Карипис и Кумар (1998) и др.
 - Геометрическое разделение (geometric partitioning) – Симон (1991), Миллер, Тенг, Вавасис (1991) и др.
 - Спектральное разделение (spectral partitioning) – Донат и Хоффман (1973), Фидлер (1975), Барнард и Симон (1993) и др.
 - **Многоуровневое разделение (multilevel graph partitioning)** – Хендриксон и Леланд (1993), Карипис и Кумар (1996) и др.

Растущее разделение графа

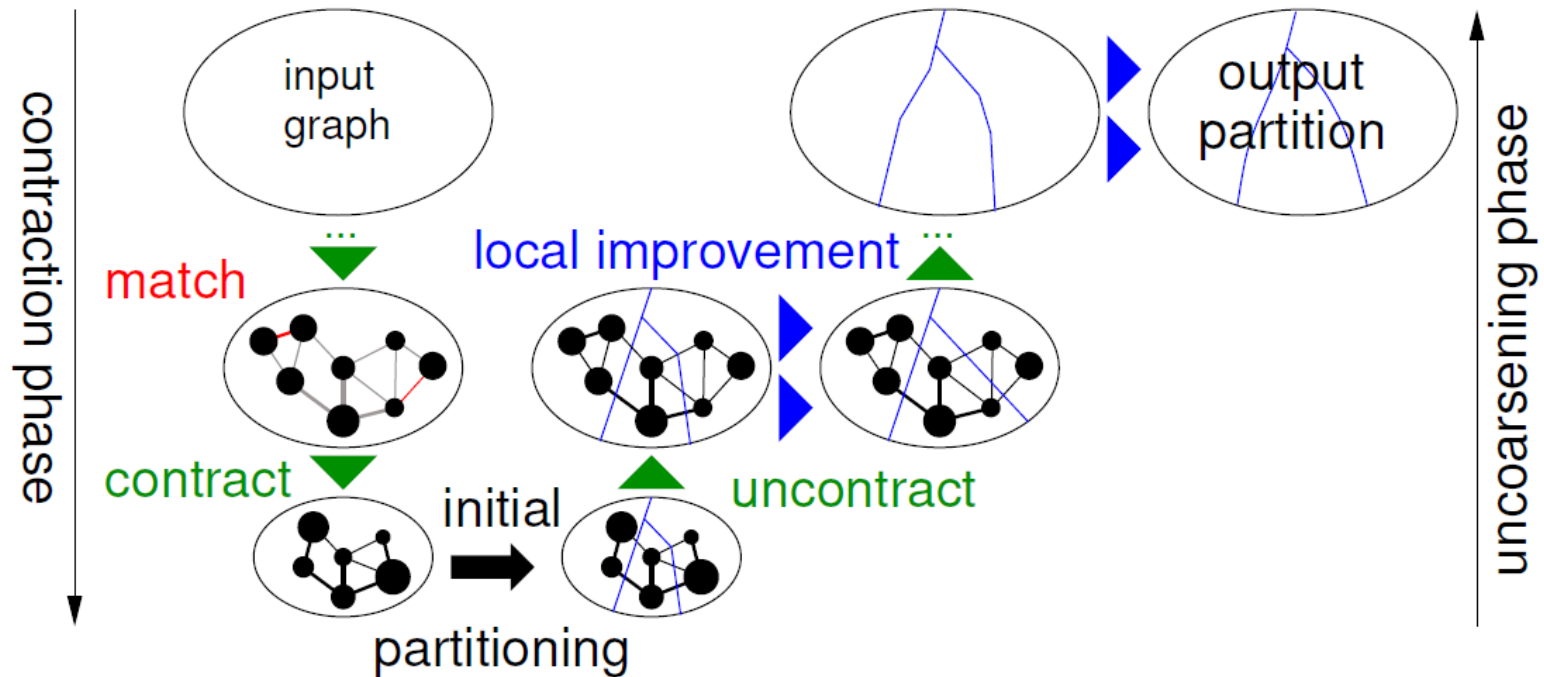
- ❑ Идея: «выращиваем» части разделения поиском в ширину из разных начальных вершин.
- ❑ Бисекция графа:
 - в часть V_1 включаем вершины, достигнутые поиском в ширину из случайной вершины v .
 - Процесс выполняется до тех пор, пока $|V_1| < |V|/2$. Оставшиеся вершины включаем в часть V_2 .
- ❑ Перезапуск алгоритма из разных начальных вершин существенно улучшает качество разделения

Спектральное разделение графа

- ❑ Идея: использовать свойства собственных чисел и собственных векторов матрицы Лапласа графа.
- ❑ Бисекция графа:
 - вычислить собственный вектор, соответствующий второму по величине собственному числу (вектор Фидлера). Для этого использовать модифицированный алгоритм Ланцоша.
 - Выбрать медиану вектора $x^{(l)}$. Тогда все вершины графа можно разделить на два подмножества: $V_1 = \{v: x^{(v)} \leq x^{(l)}\}, V_2 = V \setminus V_1$.
- ❑ К-разделение графа: использовать несколько собственных векторов, либо применить рекурсивную бисекцию.
- ❑ **Преимущества:** высокая точность решения, можно использовать ПО для линейной алгебры
- ❑ **Недостаток метода:** большая вычислительная трудоемкость

Многоуровневое разделение графа

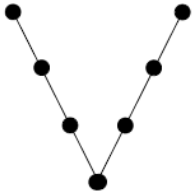
- Идея: уменьшить размерность задачи, найти разделение маленького графа, и спроецировать его на исходный большой граф



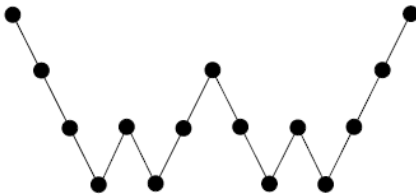
Buluç A. et al. Recent advances in graph partitioning //Algorithm Engineering. – Springer, Cham, 2016. – С. 117-158.

Многоуровневое разделение графа

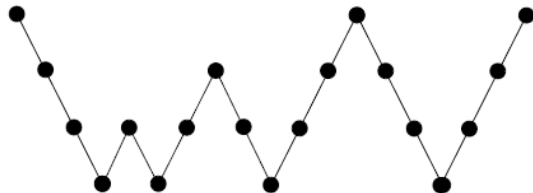
□ Варианты многоуровневой схемы:



V-Cycle



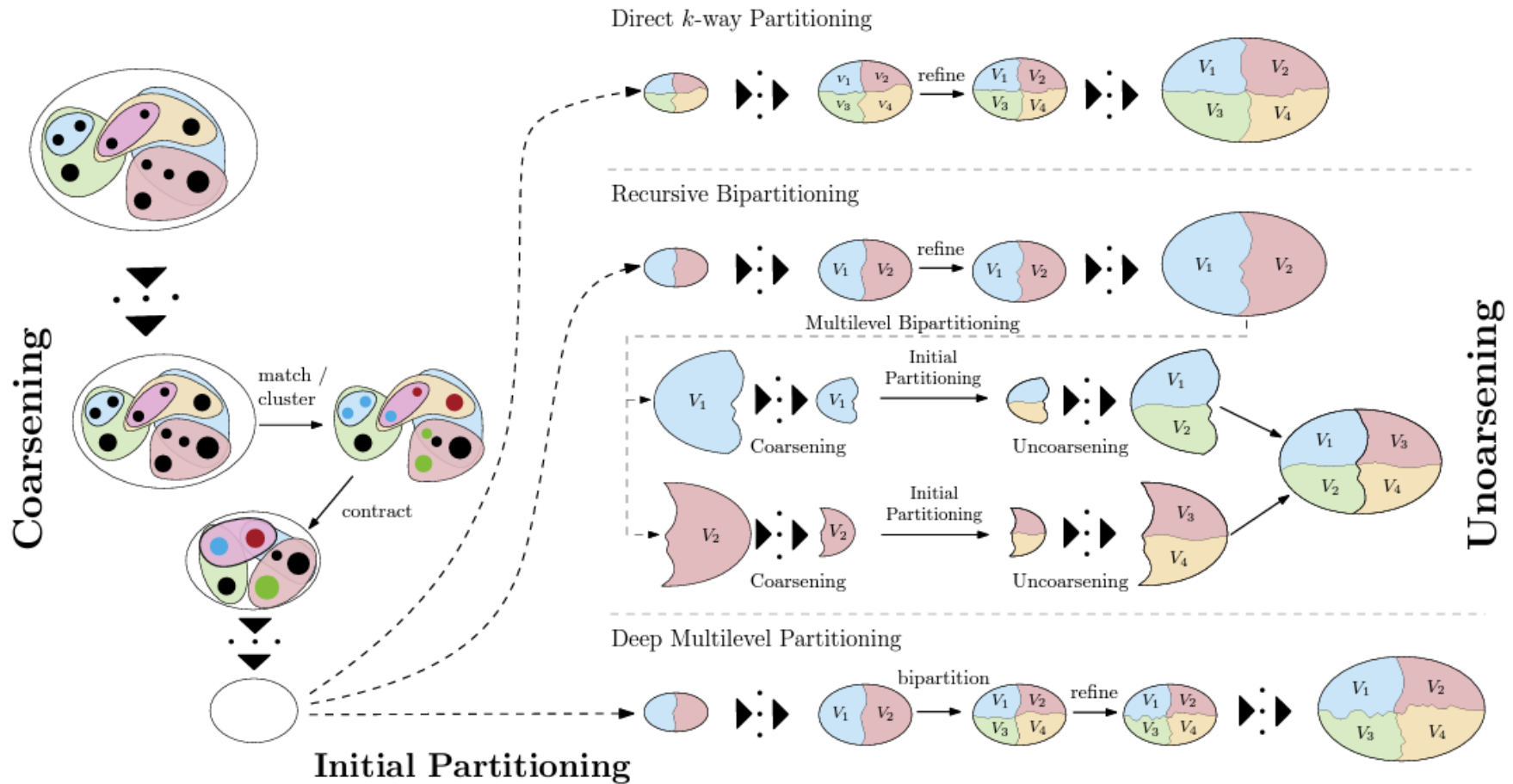
W-Cycle



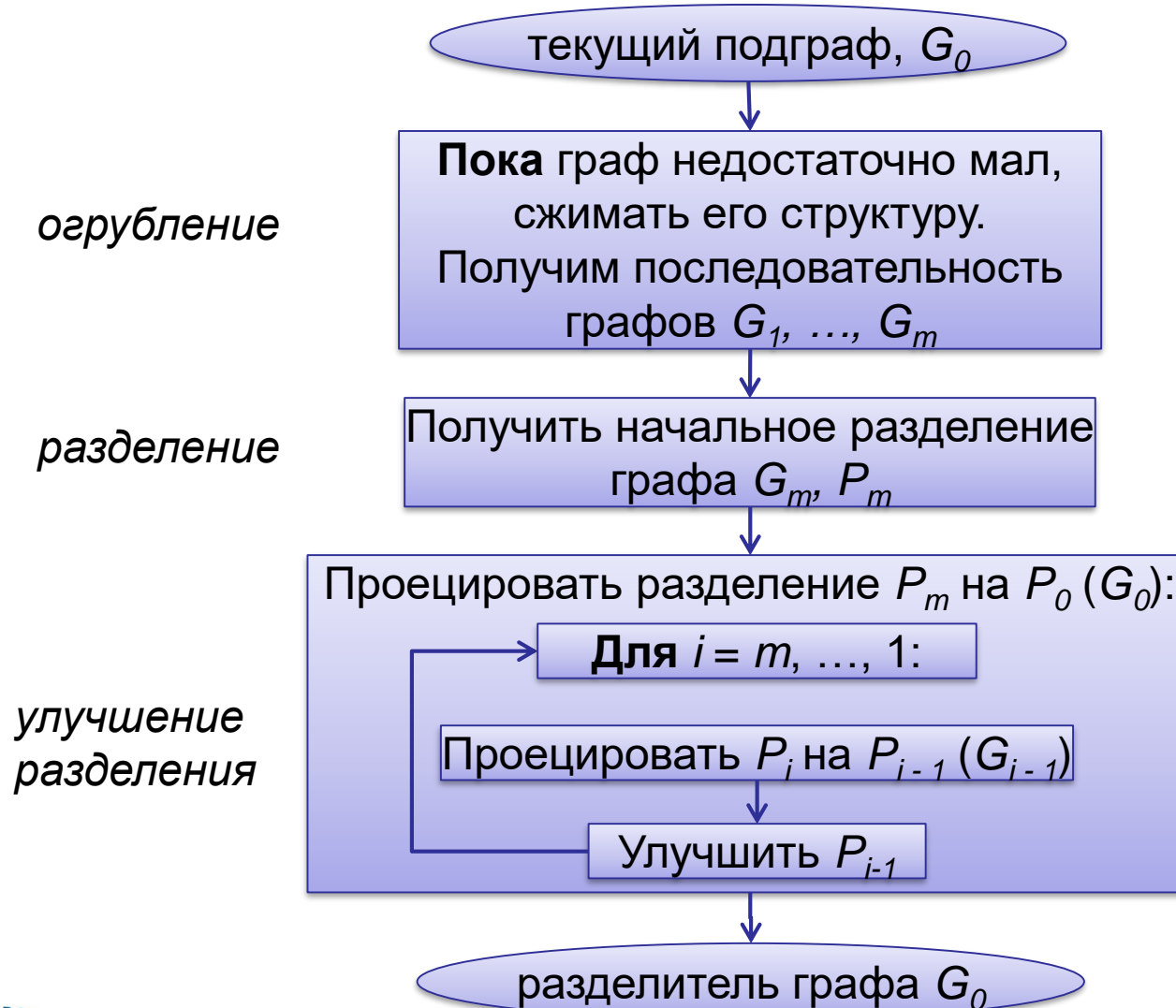
F-Cycle

Многоуровневое разделение графа

Многоуровневое k -разделение, разные схемы



Многоуровневое разделение графа



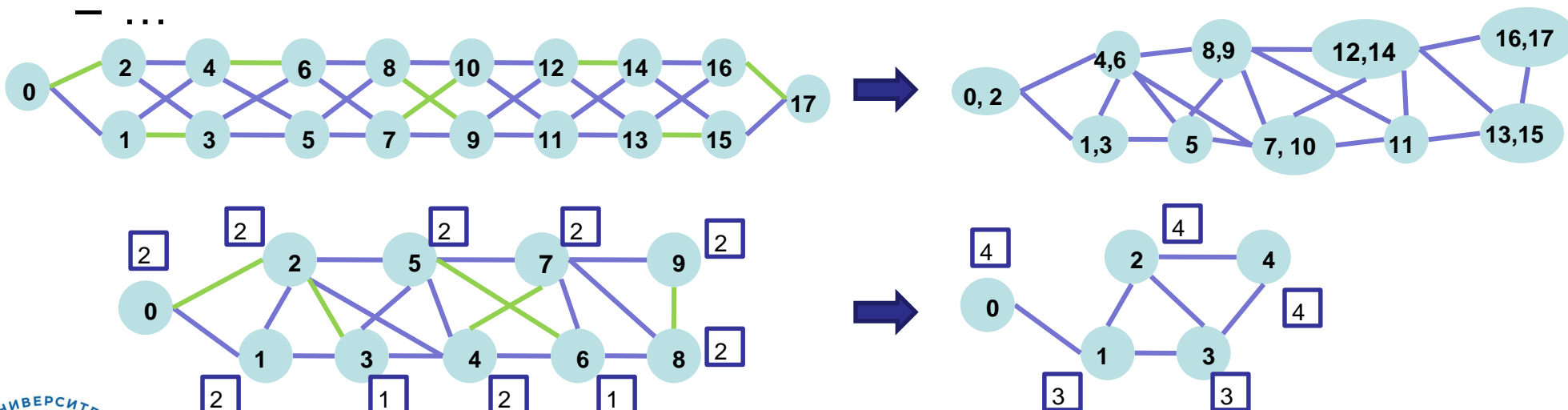
Огрубление графа (graph coarsening)

❑ **Способ 1:** Поиск максимального паросочетания

❑ *Паросочетание* – множество ребер, никакая пара которых не инцидентна одной вершине

- Случайные паросочетания
- Паросочетания тяжелых ребер
- Паросочетание тяжелых клик

– ...



Огрубление графа (graph coarsening)

- ❑ **Способ 1:** Поиск максимального паросочетания
- ❑ *Паросочетание* – множество ребер, никакая пара которых не инцидентна одной вершине
 - Случайные паросочетания
 - Паросочетания тяжелых ребер
 - Паросочетание тяжелых клик
 - ...
- ❑ Сложность $O(|V| + |E|)$
- ❑ Граф на следующей итерации имеет до 2 раз меньше вершин
- ❑ Подходит для регулярных графов (сетки, МКЭ)

Огрубление графа (graph coarsening)

□ Способ 2: алгоритмы кластеризации

– Label propagation (KaHIP)

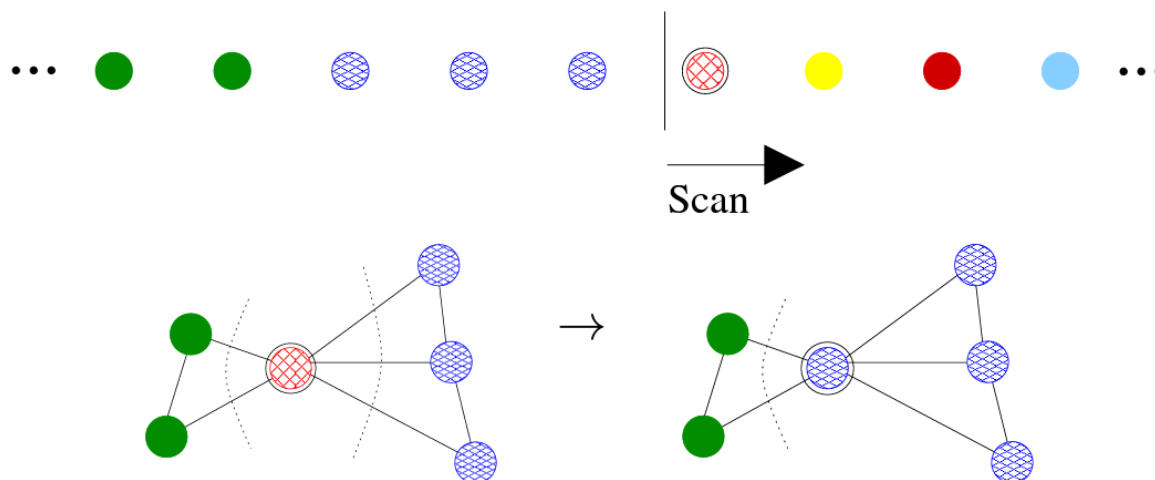


Fig. 2. An example round of the label propagation graph clustering algorithm. Initially each node is in its own block. The algorithm scans all vertices in a random order and moves a node to the block with the strongest connection in its neighborhood.

Meyerhenke H., Sanders P., Schulz C. Parallel graph partitioning for complex networks //IEEE Transactions on Parallel and Distributed Systems. – 2017. – T. 28. – №. 9. – С. 2625-2638.

Огрубление графа (graph coarsening)

- ❑ **Способ 2:** алгоритмы кластеризации
 - Label propagation (KaHIP)
- ❑ Подходит для scale-free графов
- ❑ После сжатия в новом графе значительно меньше вершин
- ❑ Сложность $O(|V| + |E|)$

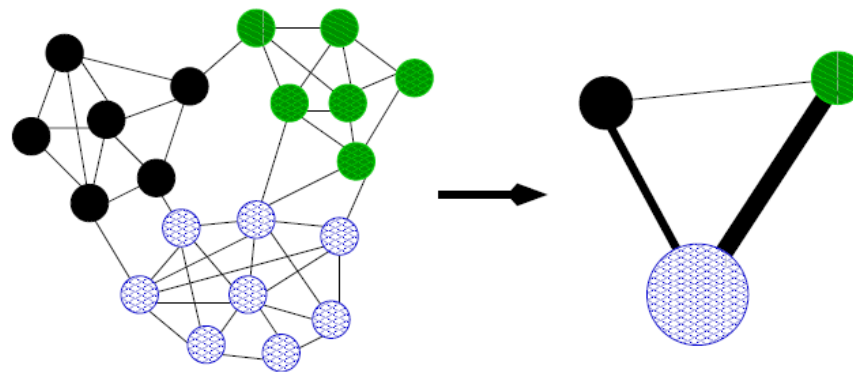


Fig. 3. Contraction of clusterings. Each cluster of the graph on the left hand side corresponds to a node in the graph on the right hand side. Weights of the nodes and the edges are chosen such that a partition of the coarse graph induces a partition of the fine graph having the same cut and balance.

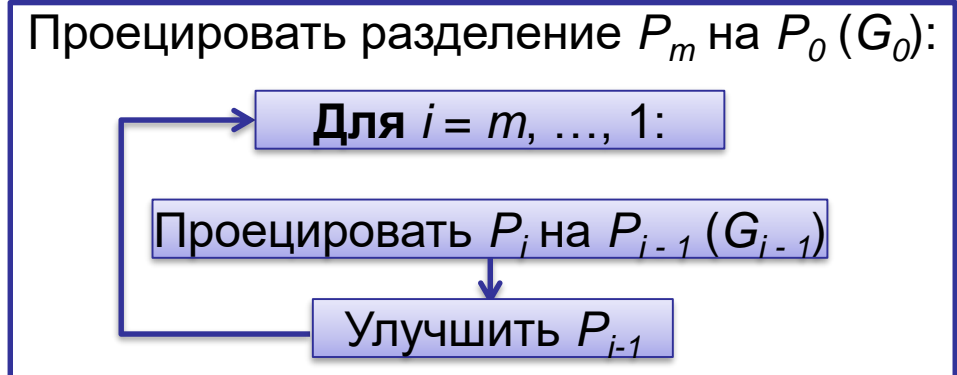
Meyerhenke H., Sanders P., Schulz C. Parallel graph partitioning for complex networks //IEEE Transactions on Parallel and Distributed Systems. – 2017. – T. 28. – №. 9. – С. 2625-2638.

Начальное разделение

- ❑ Любой алгоритм разделения графа
 - алгоритм растущего разделения, запущенный из разных начальных вершин
 - алгоритм жадного растущего разделения...
- ❑ Сложность $O(|V| + |E|)$

Улучшение разделения

- ❑ Алгоритмы локальной дискретной оптимизации существенно улучшают качество разделения
- ❑ Методы улучшения разделения:
 - Метод Кернигана–Лина (KL), его модификация – метод Федучиа – Мэтьюса (FM)
 - tabu search
 - Построение задачи минимизации разреза для потоков в сетях
 - Диффузия
 - Генетические алгоритмы
 - Label propagation



Метод Кернигана–Лина / Федучча–Мэтьюса

- ❑ **Идея:** перемещать по одной вершине из большей части разделения в меньшие, согласно приоритету (FM) ИЛИ обменивать пары вершин между частями разделения (KL)
- ❑ Приоритет – оценка уменьшения разреза
- ❑ Пусть граф разделен на две части, V_1, V_2 . Введем функцию, которая для каждой граничной вершины $v \in V_2$ оценивает уменьшение разреза графа, если включить v в V_1 . Как правило, это разница суммы весов «исходящих» ребер и «внутренних» ребер:

$$gain(v) = \sum_{u \in V_1} w(v, u) - \sum_{z \in V_2} w(v, z)$$

Метод Кернигана–Лина / Федучча–Мэтьюса

❑ Использование приоритетов:

- Граничные вершины упорядочим в приоритетной очереди по значению $gain$
- При поиске в ширину в часть V_1 включаем вершину $v \in V_2$ с максимальным значением $gain(v)$
- После каждого изменения части V_1 требуется пересчет $gain$ для соседних с v вершин

❑ Способы хранения gains: карманы (buckets)

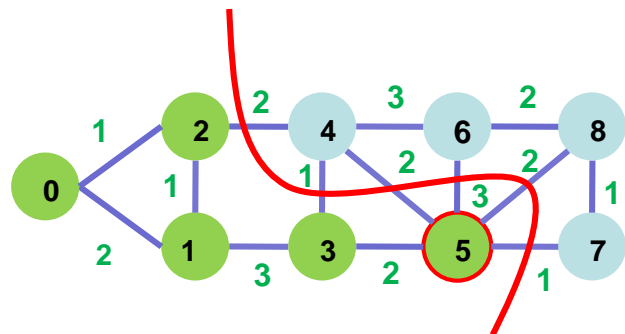
❑ К-разделение: k приоритетных очередей либо одна очередь с модифицированным $gain$

❑ Нужен критерий сравнения качества разделений

Метод Кернигана–Лина / Федучча – Мэтьюса

- Вход: стартовое разделение $P_s = (V_1, V_2)$
- Выход: улучшенное разделение $P^{new} = (V_1^{new}, V_2^{new})$
 1. Определить «меньшую» часть V_1 , «большую» часть V_2
 2. Для граничных вершин вычислить $gain$ перемещения в V_1
 3. **Пока** (P^{new} лучше P_s) И (есть непосещенные вершины):
 1. Выбрать вершину $u \in V_2$ с максимальным $gain$ (ИЛИ пару для обмена)
 2. Переместить u : $V_1 = V_1 \cup \{u\}$, $V_2 = V_2 / \{u\}$. Пометить u посещенной
 3. Пересчитать $gain$ для соседних с u вершин
- Сложность зависит от способа хранения gains.
 - Оригинальный алгоритм KL - $O(|V|^2 \log |V|)$
 - Модификация FM - $O(|E|)$ (карманы)

Метод Кернигана–Лина (Федуччия – Мэтьюса)



$$edgeCut = 11$$

$$Gain(2) = 2 - 1 - 1 = 0$$

$$Gain(3) = 1 - 3 - 2 = -4$$

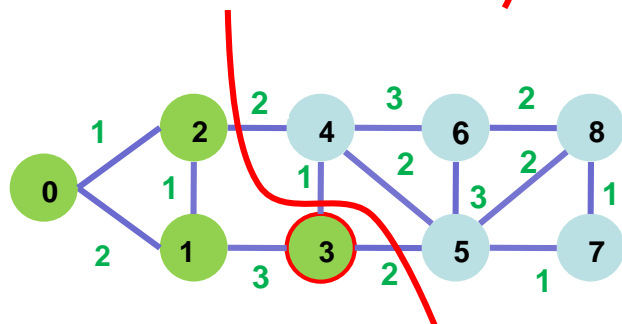
$$Gain(5) = 2 + 3 + 2 + 1 - 2 = 6$$

$$Gain(4) = 5 - 3 = 2$$

$$Gain(6) = 3 - 5 = -2$$

$$Gain(7) = 1 - 1 = 0$$

$$Gain(8) = 2 - 3 = -1$$



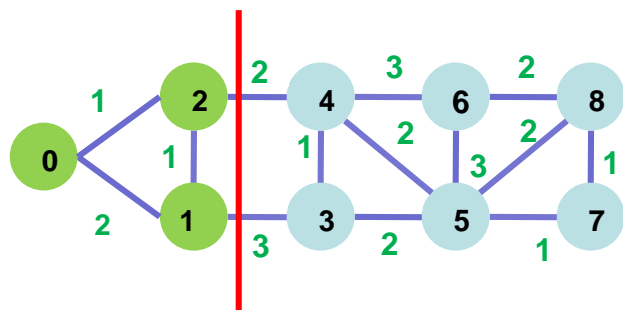
$$edgeCut = 5$$

$$Gain(2) = -2$$

$$Gain(3) = 3 - 3 = 0$$

$$Gain(4) = 3 - 5 = -2$$

$$Gain(5) = 2 - 8 = -6$$



$$edgeCut = 5$$

$$Gain(1) = -2$$

$$Gain(2) = 2 - 2 = 0$$

$$Gain(3) = 3 - 3 = 0$$

$$Gain(4) = 2 - 6 = -4$$

Библиотеки для разделения графов

- ❑ Большинство существующих параллельных библиотек для разделения графов предназначены для систем с распределенной памятью. В 2013 г. впервые была опубликована библиотека для систем с общей памятью (mt-metis)
- ❑ Популярные библиотеки, в которых реализован многоуровневый метод:
 - [Metis, ParMETIS](#) (с 1996 г.), mt-metis (2013)
 - [Scotch, PT-Scotch](#) (с 1996 г.)
 - [KaHIP](#) (2011-2020)
- ❑ Другие библиотеки:
 - [Jostle](#), Party, DibaP, PDibaP, Parkway, [Zoltan](#)

Параллельный алгоритм

- ❑ Особенности распараллеливания разделения графов:
 - Чтобы качество разделения параллельной версии алгоритма было близким к результатам работы последовательной версии, требуется большое число синхронизаций между потоками (процессами).
 - На практике используются модификации последовательных алгоритмов, дающие достаточное качество разделения за приемлемое время.
 - При увеличении числа процессов (потоков) качество разделения, как правило, снижается.

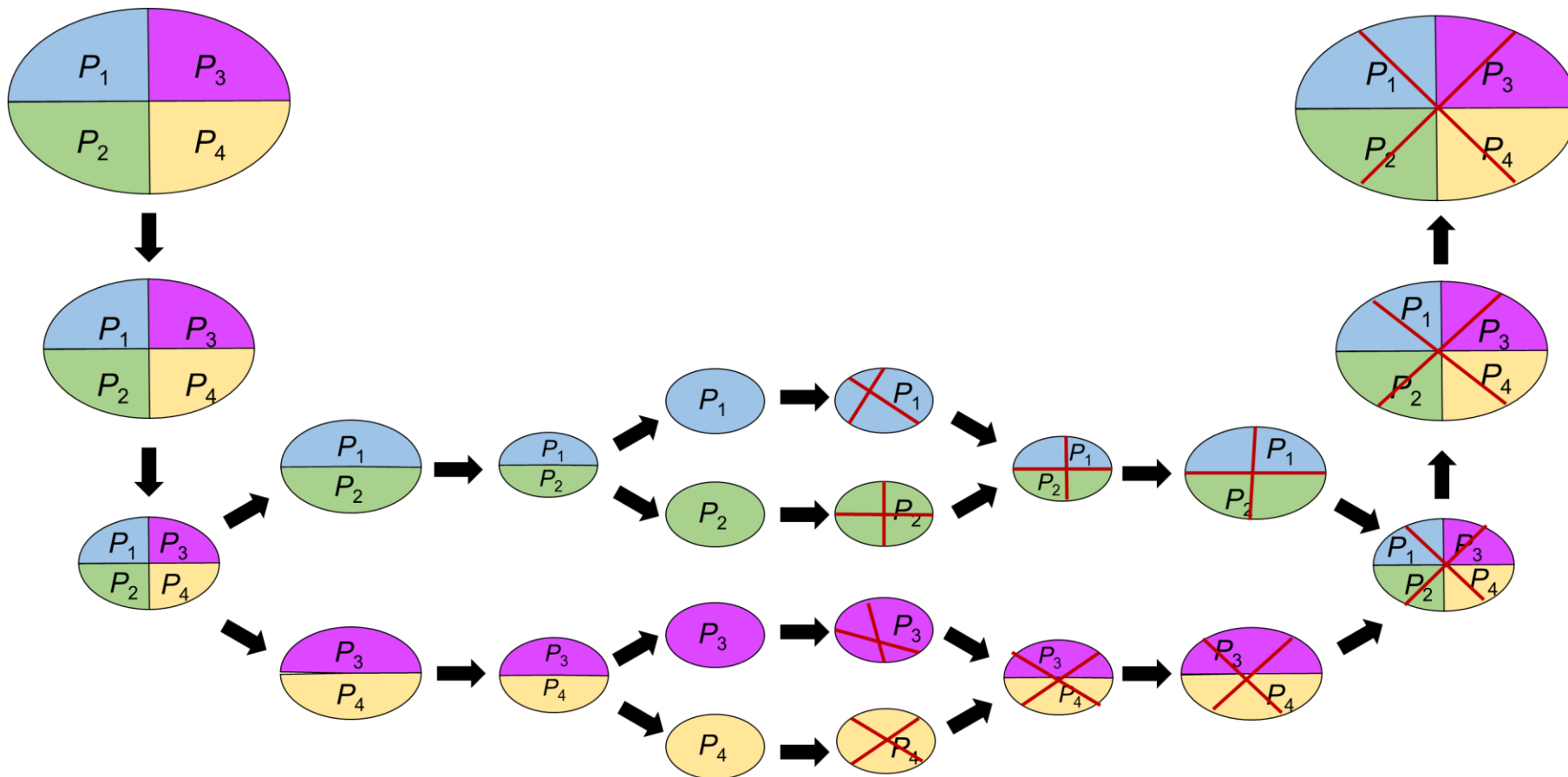
Параллельный алгоритм

- ❑ Способы разделения исходного графа:
 - 1D, «полосами» - PT-Scotch
 - 2D, блоками – ParMETIS

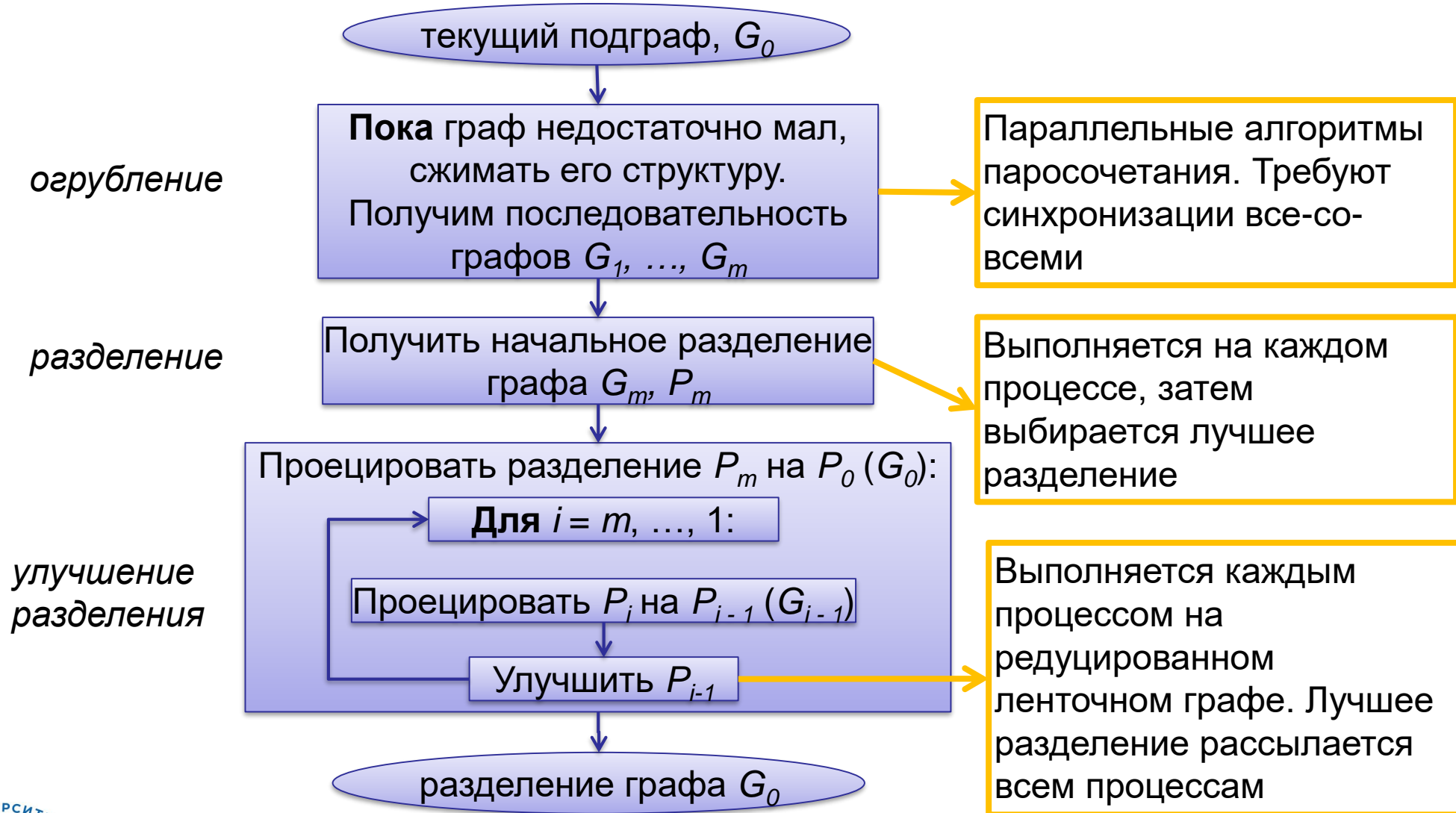
- ❑ Перед началом работы алгоритма граф как-то распределен по процессам → нужна предобработка, перераспределение вершин
 - Раскраска графа (ParMETIS)

Параллельный алгоритм (PT-Scotch)

□ «Сложение и дублирование»



Параллельный алгоритм

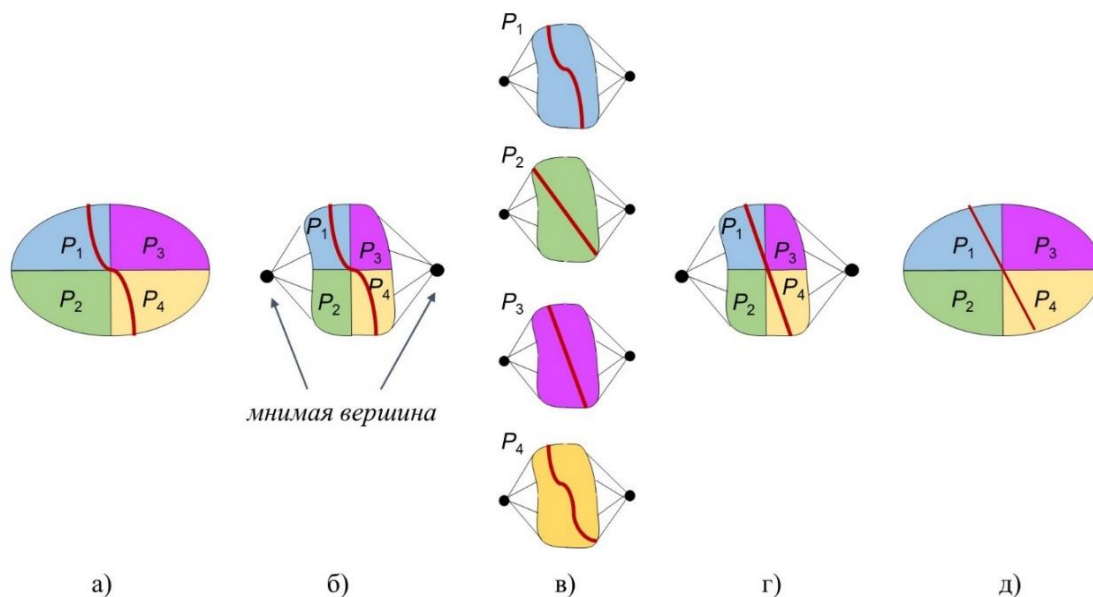


Метод Кернигана–Лина (Федучча–Мэтьюса)

□ Подходы к распараллеливанию:

- Вариант 1. Одновременная обработка приоритетной очереди разными процессами (потоками). Каждый из них перемещает только свою группу вершин.
- Вариант 2. На каждом процессе (потоке) выполнять улучшение разделения для фрагмента графа, затем выбрать лучшее разделение.

В качестве фрагмента рассматривается часть графа вокруг граничных вершин.



Бенчмарки

- ❑ [The Walshaw benchmark](#) - один из популярных бенчмарков. Это коллекция из 34 графов из разных прикладных областей. Для каждого графа содержит разделения на 2, 4, 8, 16, 32, 64 части с наименьшим найденным разрезом. При этом время работы реализаций не сравнивается.
- ❑ [10th DIMACS Implementation Challenge](#)
- ❑ Сравнение времени работы решателей СЛАУ на разделенном графе

Результаты вычислительных экспериментов

- ❑ Сравнение реализаций для общей и распределенной памяти PT-Scotch, ParMETIS и mt-metis (2015 г.)

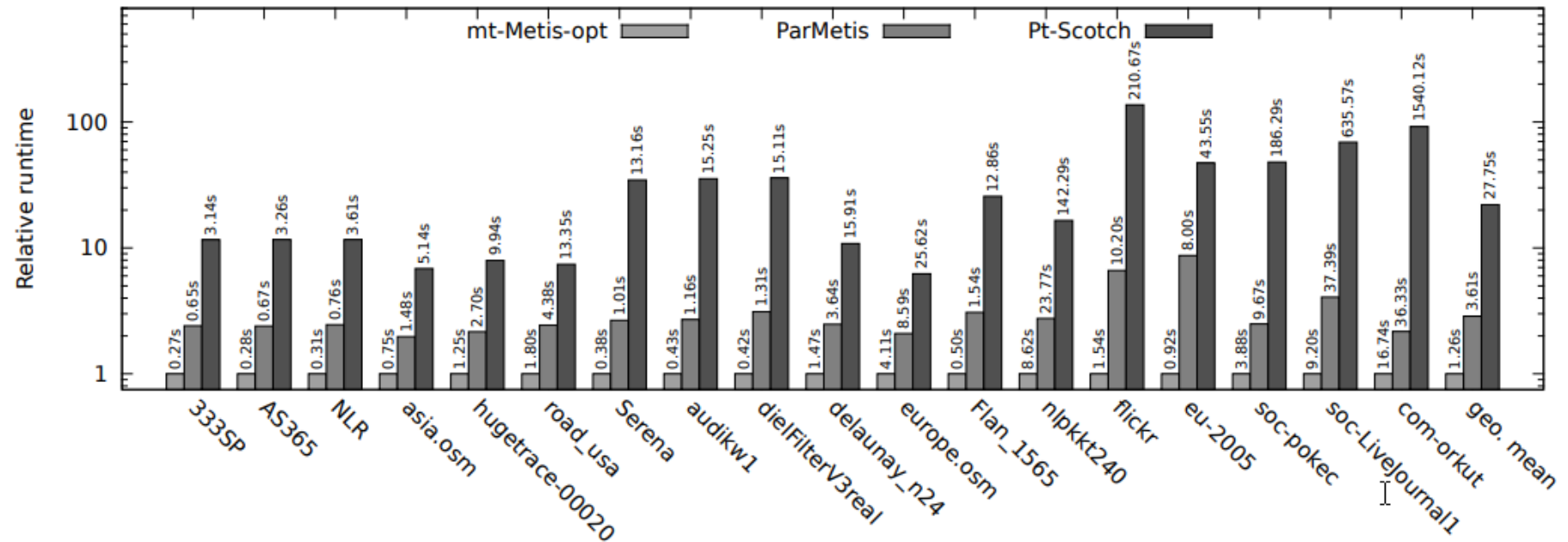


Figure 9: Comparison of modified mt-Metis with other partitioners, using 36 threads/processes and $k = 64$. Runtimes are relative with respect to the runtime of mt-Metis-opt. Absolute runtimes in seconds are shown above the corresponding bars.

Результаты вычислительных экспериментов

- Сравнение реализаций для распределенной памяти KaHIP и ParMETIS (2015 г.)

TABLE I. BASIC PROPERTIES OF THE BENCHMARK SET WITH A ROUGH TYPE CLASSIFICATION. S STANDS FOR SOCIAL OR WEB GRAPHS, M IS USED FOR MESH TYPE NETWORKS.

graph	n	m	Type	Ref.
Large Graphs				
amazon	$\approx 407K$	$\approx 2.3M$	S	[39]
eu-2005	$\approx 862K$	$\approx 16.1M$	S	[36]
youtube	$\approx 1.1M$	$\approx 2.9M$	S	[39]
in-2004	$\approx 1.3M$	$\approx 13.6M$	S	[36]
packing	$\approx 2.1M$	$\approx 17.4M$	M	[36]
enwiki	$\approx 4.2M$	$\approx 91.9M$	S	[41]
channel	$\approx 4.8M$	$\approx 42.6M$	M	[36]
hugebubble-10	$\approx 18.3M$	$\approx 27.5M$	M	[36]
nlpkkt240	$\approx 27.9M$	$\approx 373M$	M	[37]
uk-2002	$\approx 18.5M$	$\approx 262M$	S	[41]
del26	$\approx 67.1M$	$\approx 201M$	M	[23]
rgg26	$\approx 67.1M$	$\approx 575M$	M	[23]
Larger Web Graphs				
arabic-2005	$\approx 22.7M$	$\approx 553M$	S	[41]
sk-2005	$\approx 50.6M$	$\approx 1.8G$	S	[41]
uk-2007	$\approx 105.8M$	$\approx 3.3G$	S	[41]
Graph Families				
delX	$[2^{19}, \dots, 2^{31}]$	$\approx 1.5M-6.4G$	M	[23]
rggX	$[2^{19}, \dots, 2^{31}]$	$\approx 3.3M-21.9G$	M	[23]

Инфраструктура:
four Intel Xeon E5-4640
Octa-Core processors
(Sandy Bridge) running
at a clock speed of 2.4
GHz. The machine has
512 GB main memory,
20 MB L3-Cache and
8x256 KB L2-Cache

Результаты вычислительных экспериментов

□ Сравнение реализаций для распределенной памяти KaHIP и ParMETIS (2015 г.)

TABLE II. AVERAGE PERFORMANCE (CUT AND RUNNING TIME) AND BEST RESULT ACHIEVED BY DIFFERENT PARTITIONING ALGORITHMS. RESULTS ARE FOR THE BIPARTITIONING CASE $k = 2$. ALL TOOLS USED 32 PEs OF MACHINE A. RESULTS INDICATED BY A * MEAN THAT THE AMOUNT OF MEMORY NEEDED BY THE PARTITIONER EXCEEDED THE AMOUNT OF MEMORY AVAILABLE ON THAT MACHINE WHEN 32 PEs ARE USED (512GB RAM). THE PARMETIS RESULT ON ARABIC HAS BEEN OBTAINED USING 15 PEs (THE LARGEST NUMBER OF PEs SO THAT PARMETIS COULD SOLVE THE INSTANCE).

algorithm	ParMetis			Fast			Eco		
graph	avg. cut	best cut	$t[s]$	avg. cut	best cut	$t[s]$	avg. cut	best cut	$t[s]$
amazon	48 104	47 010	0.49	46 641	45 872	1.85	44 703	44 279	71.04
eu-2005	33 789	24 336	30.60	20 898	18 404	1.63	18 565	18 347	70.04
youtube	181 885	171 857	6.10	174 911	171 549	8.74	167 874	164 095	105.87
in-2004	7 016	5 276	3.43	3 172	3 110	1.38	3 027	2 968	69.19
packing	11 991	11 476	0.24	10 185	9 925	1.84	9 634	9 351	68.69
enwiki	9 578 551	9 553 051	326.92	9 622 745	9 565 648	157.32	9 559 782	9 536 520	264.64
channel	48 798	47 776	0.55	56 982	55 959	2.71	52 101	50 210	71.95
hugebubbles	1 922	1 854	4.66	1 918	1 857	38.00	1 678	1 620	216.91
nlpkt240	1 178 988	1 152 935	15.97	1 241 950	1 228 086	35.06	1 193 016	1 181 214	192.78
uk-2002	787 391	697 767	128.71	434 227	390 182	19.62	415 120	381 464	146.77
del26	18 086	17 609	23.74	17 002	16 703	165.02	15 826	15 690	697.43
rgg26	44 747	42 739	8.37	38 371	37 676	55.91	34 530	34 022	263.81
arabic-2005	*1 078 415	*968 871	*1 245.57	551 778	471 141	33.45	511 316	475 140	184.01
sk-2005	*	*	*	3 775 369	3 204 125	471.16	3 265 412	2 904 521	1 688.63
uk-2007	*	*	*	1 053 973	1 032 000	169.96	1 010 908	981 654	723.42

Результаты вычислительных экспериментов

❑ Слабая масштабируемость

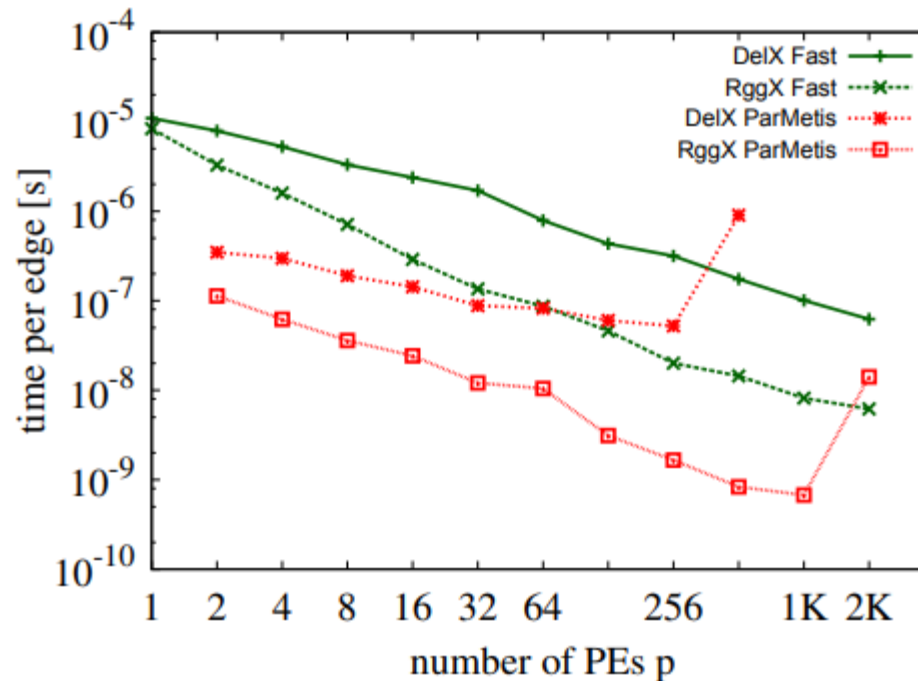
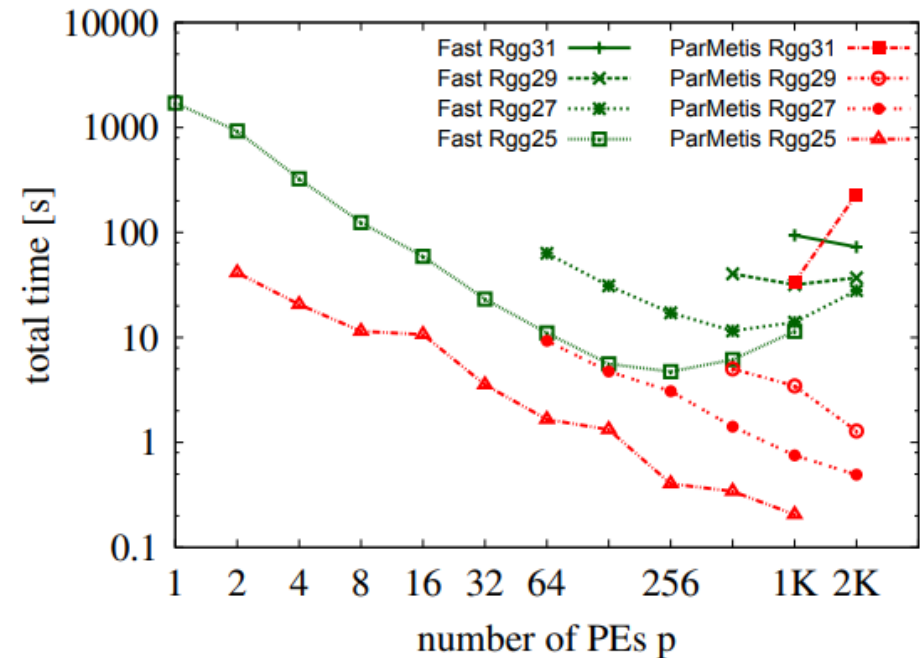
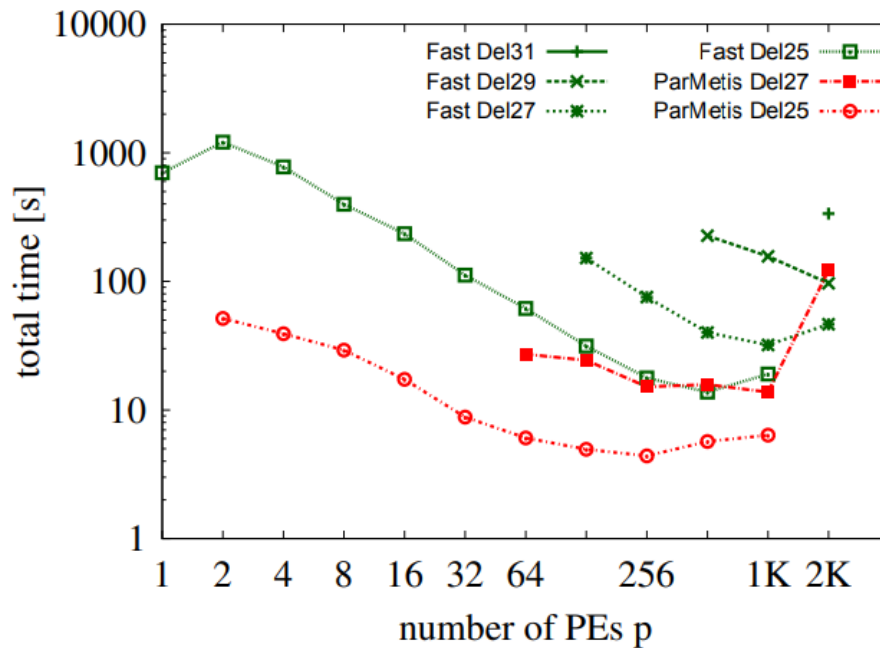


Fig. 5. Weak scaling experiments for random geometric graph class rggX and the Delaunay triangulation graph class delX. When using p PEs, the instance with $2^{19}p$ nodes from the corresponding graph class was used, i.e. when using 2048 cores all algorithms partition the graphs del30 and rgg30. The figure shows the time spend per edge. Sixteen blocks have been used for the partitioning task.

Результаты вычислительных экспериментов

□ Сильная масштабируемость



Результаты вычислительных экспериментов.

Выводы

- ❑ Выводы из результатов экспериментов для библиотек KaHIP, ParMETIS, PT_Scotch, mt-metis:
 - Близкие по качеству разделения можно получить за существенно разное время. На практике важно найти компромисс между требуемым качеством разделения и временем работы реализации
 - На модельных графах реализации для систем с распределенной памятью масштабируются вплоть до 1024 процессов.
 - В рамках одного узла многопоточная реализация mt-metis работает быстрее, чем MPI-версии, в среднем в 2 раза.

Заключение

- ❑ Задача разделения графов возникает в большом числе научных приложений. Одно из важнейших применений разделения графов – распределение данных между процессами для дальнейшей параллельной обработки.
- ❑ Разработан ряд методов для решения этой задачи. В настоящее время широко применяется многоуровневый метод разделения. Другие методы разделения можно использовать для получения начального разделения графа в рамках многоуровневой схемы.
- ❑ Для улучшения качества разделения графа используются итерационные алгоритмы локальной дискретной оптимизации.

Литература

1. Buluç A. et al. Recent advances in graph partitioning //Algorithm Engineering. – Springer, Cham, 2016. – С. 117-158.
2. Bichot C. E., Siarry P. (ed.). Graph partitioning. – ISTE, 2011.
3. Karypis G., Kumar V. A fast and high quality multilevel scheme for partitioning irregular graphs //SIAM Journal on scientific Computing. – 1998. – Т. 20. – №. 1. – С. 359-392.
4. LaSalle D., Karypis G. Multi-threaded graph partitioning //2013 IEEE 27th International Symposium on Parallel and Distributed Processing. – IEEE, 2013. – С. 225-236.
5. Meyerhenke H., Sanders P., Schulz C. Parallel graph partitioning for complex networks //IEEE Transactions on Parallel and Distributed Systems. – 2017. – Т. 28. – №. 9. – С. 2625-2638.
6. LaSalle D. et al. Improving graph partitioning for modern graphs and architectures //Proceedings of the 5th Workshop on Irregular Applications: Architectures and Algorithms. – ACM, 2015. – С. 14.

Контакты

Нижегородский государственный университет

<http://www.unn.ru>

Институт информационных технологий, математики и механики

<http://www.itmm.unn.ru>

Пирова А.Ю.

anna.pirova@itmm.unn.ru