


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 1. Введение

Пирова А.Ю.
Кафедра МОСТ

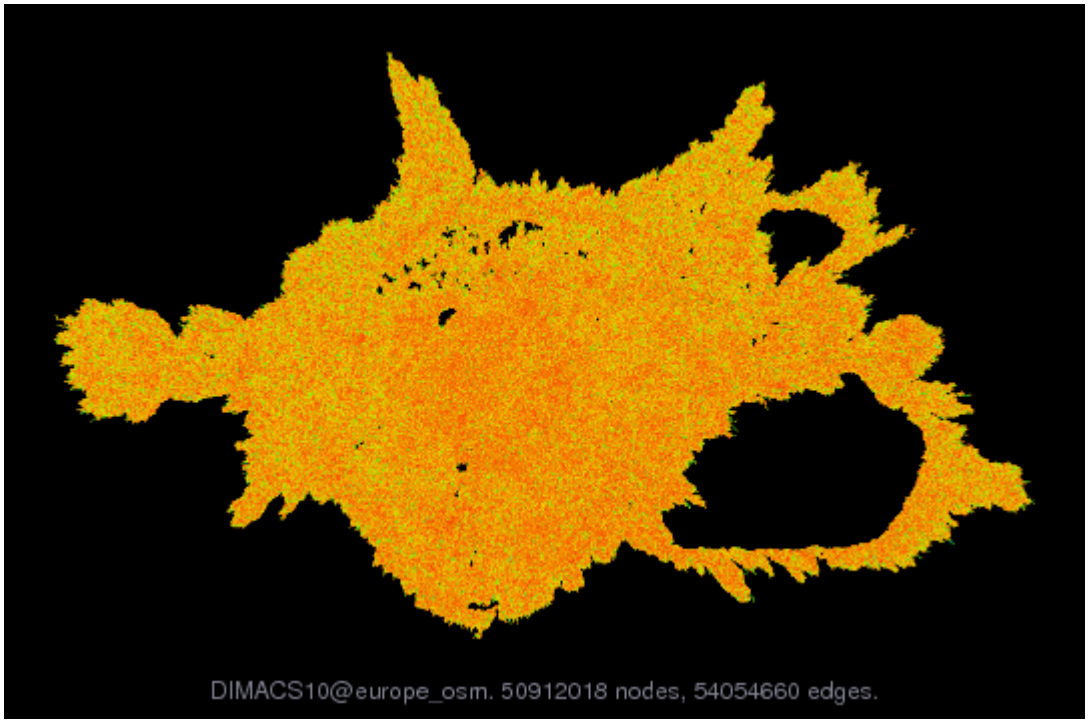
Содержание

- Прикладные области
- Математические определения
- Основные задачи на графах
- Особенности параллельных алгоритмов на графах
- Форматы хранения графов

Прикладные области

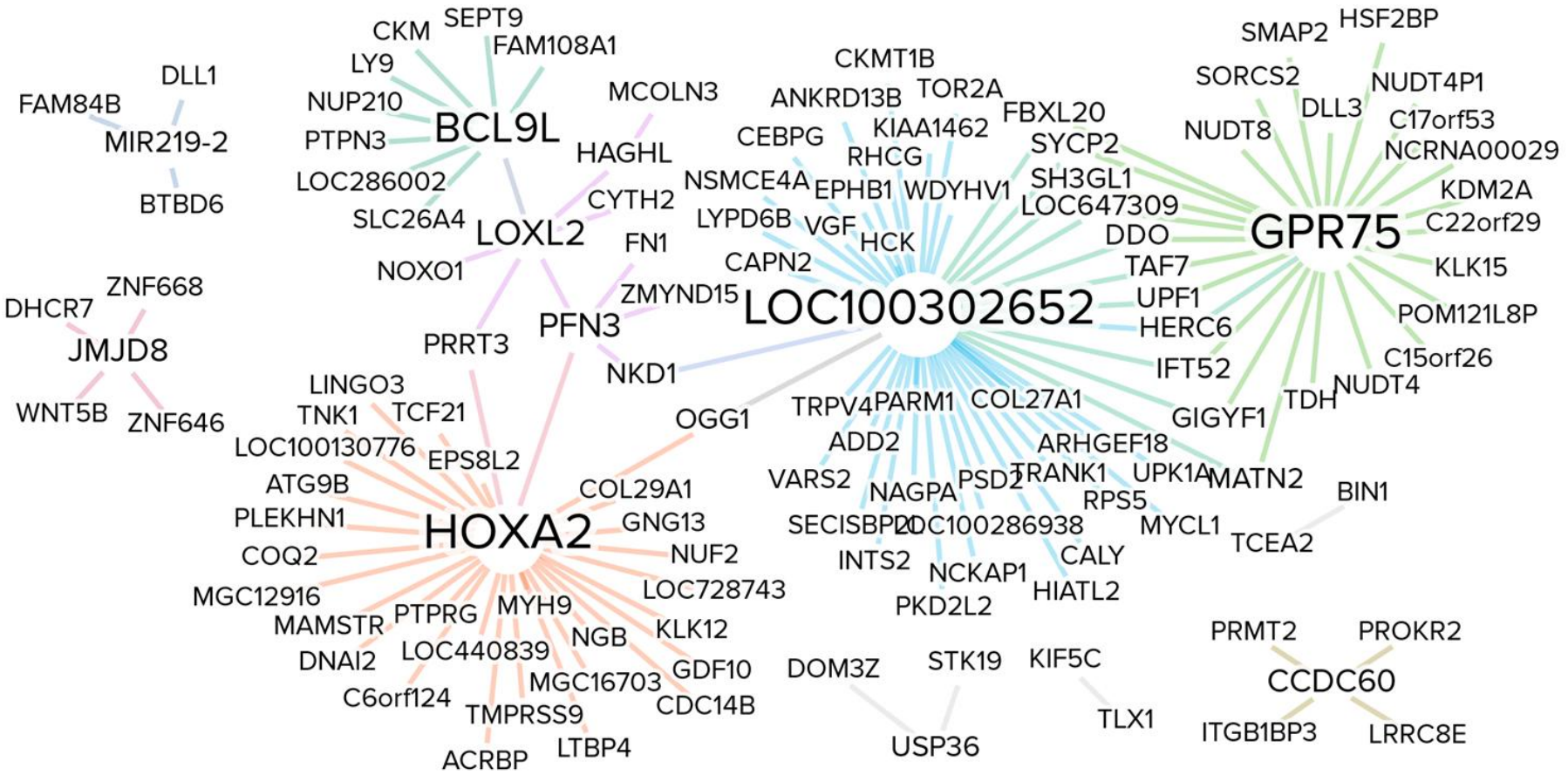
- Где возникают графы большого порядка?
  - Транспортные сети
  - Коммуникационные сети
  - Веб-графы, сети цитирований
  - Социальные сети, сети сообществ
  - Энергетические системы
  - Биологические системы (взаимодействие между белками, метаболические, нейронные сети, пищевые цепочки)
  - Бизнес-процессы, кибербезопасность

Примеры



Граф сети дорог Европы
https://sparse.tamu.edu/DIMACS10/europe_osm

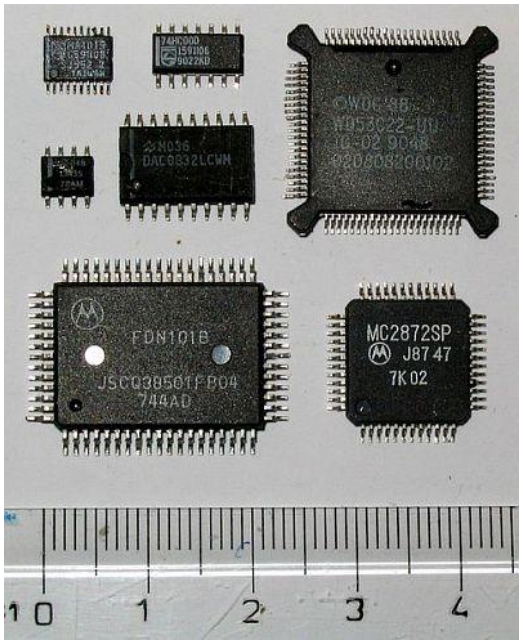
Примеры



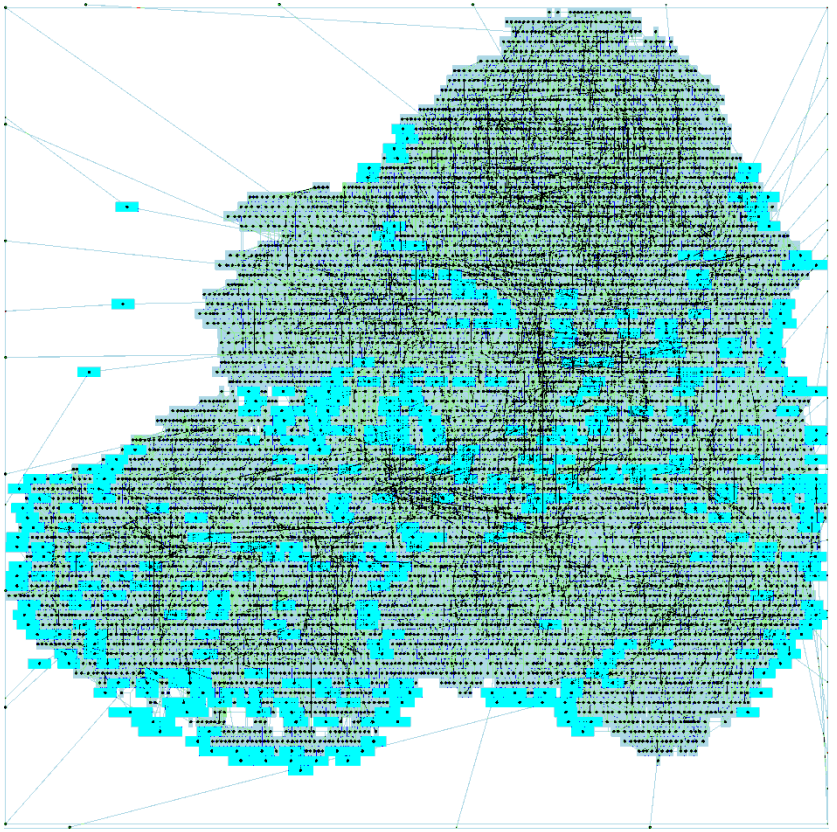
Граф корреляции в сети астроцитов. Вершина — астроцит, ребро означает значимую корреляцию между кальциевой активностью пары астроцитов. (Автор: М. Кривоносов)

Примеры

Настоящие интегральные схемы



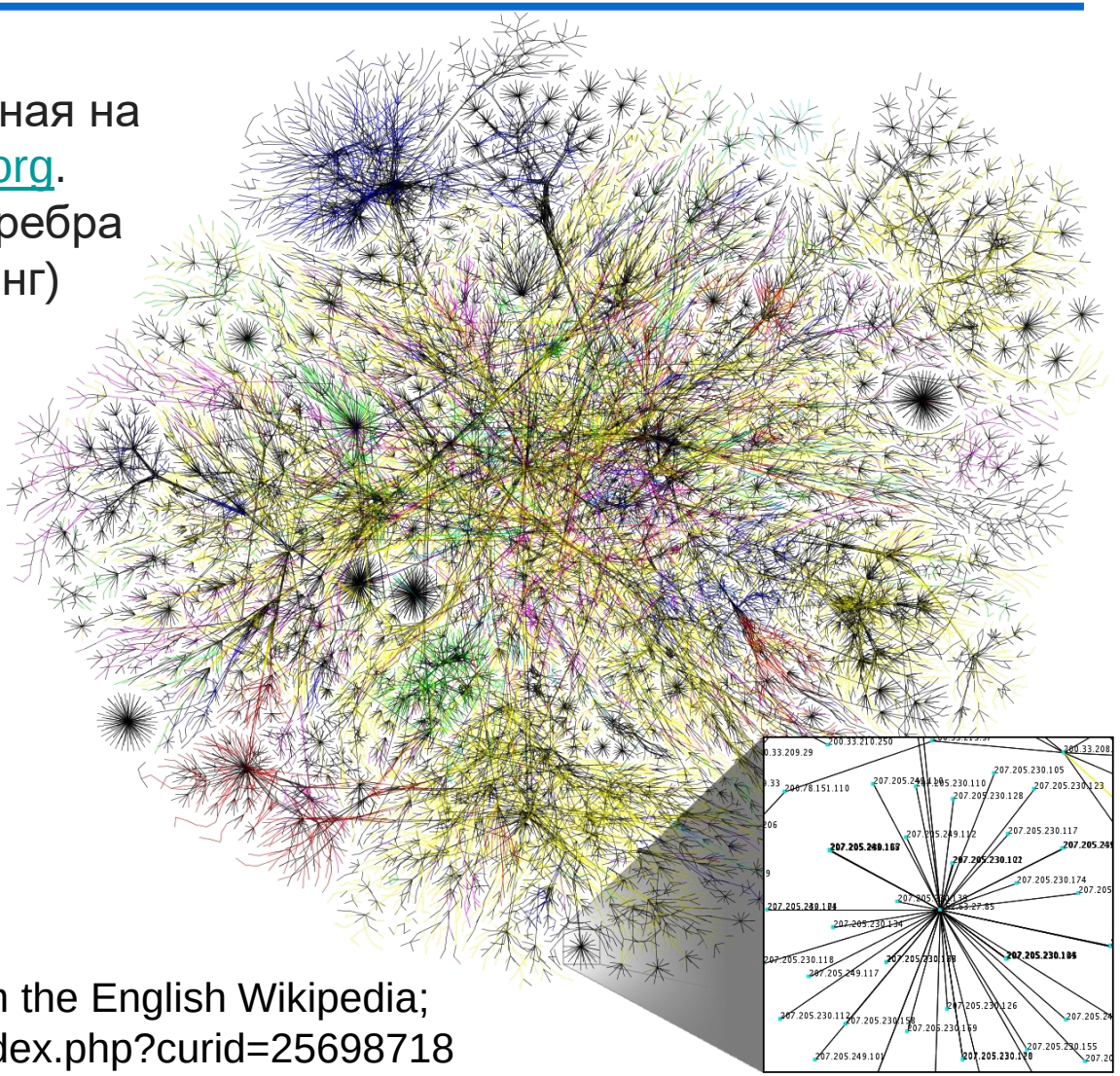
Модель интегральной схемы. Вершины графа - элементы схемы, гиперребра – их соединения



Автор: И. Лебедев

Примеры

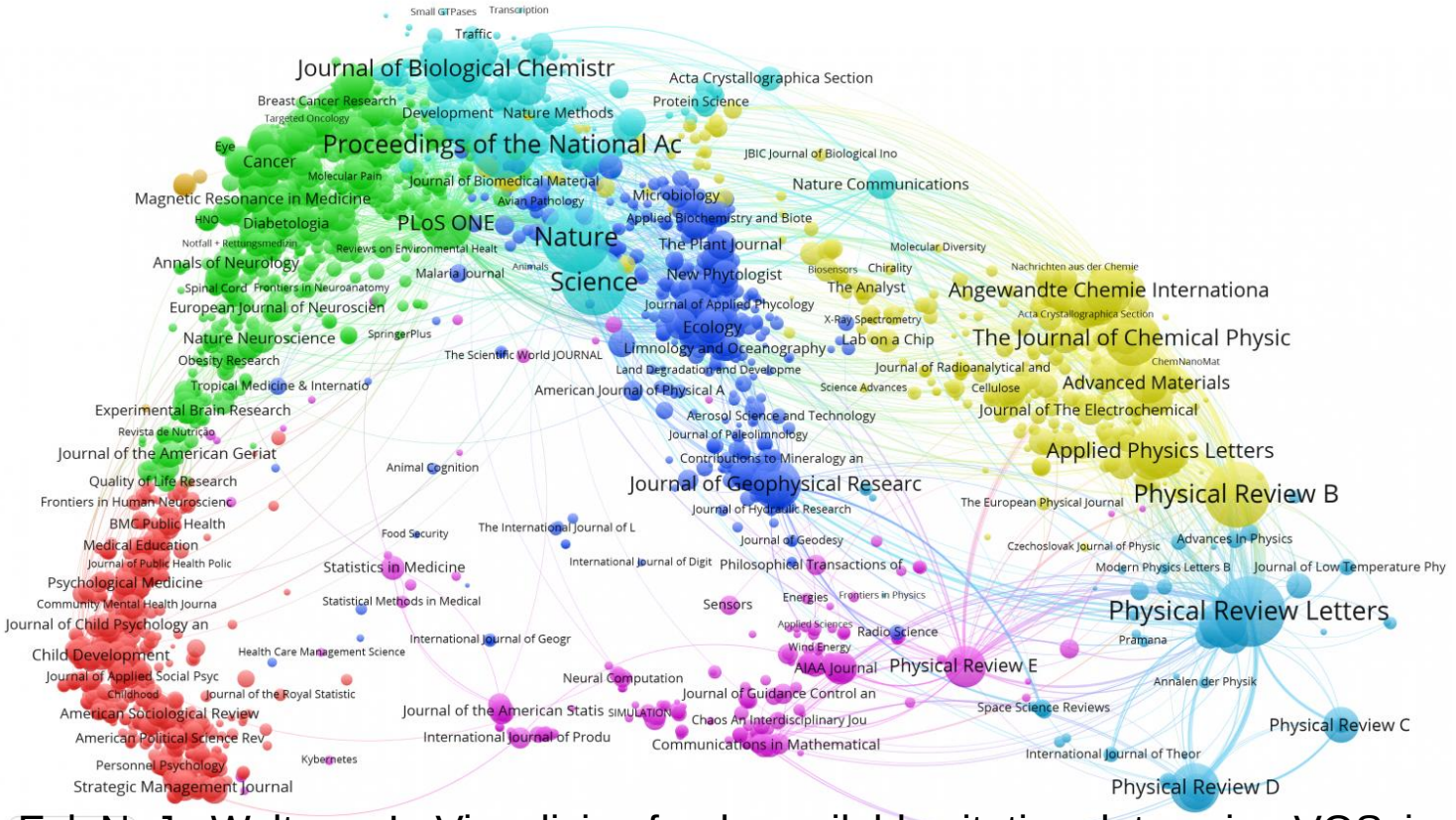
Частичная карта интернета, основанная на данных от 15 января 2005 г. на opte.org. Вершины графа – IP-адреса. Длина ребра показывает временную задержку (пинг) между двумя узлами.



By The Opte Project - Originally from the English Wikipedia; https://commons.wikimedia.org/w/index.php?curid=25698718

Примеры

Граф цитат публикаций в научных журналах за 1980-2016 гг. Вершины графа – журналы, ребра – цитирования публикаций. Визуализация с помощью программы VOSviewer, на ней отображено 5000 наиболее цитируемых журналов.



Van Eck N. J., Waltman L. Visualizing freely available citation data using VOSviewer //CWTS. Retrieved. – 2017. – Т. 9. https://www.cwts.nl/blog?article=n-r2r294

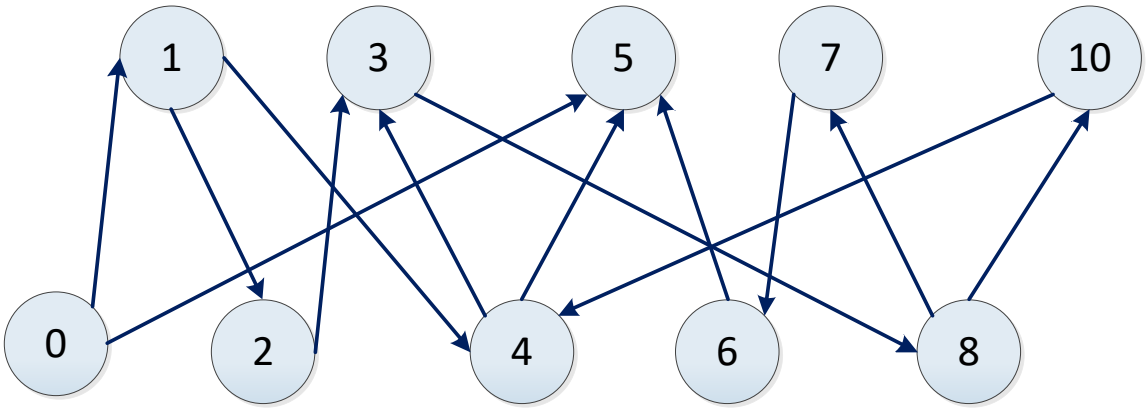
МАТЕМАТИЧЕСКИЕ ОПРЕДЕЛЕНИЯ

Некоторые типы графов. Общее определение

- *Граф* $G(V, E)$ – пара множеств, где V – конечное множество, называемое *вершинами* графа, $E \subseteq V \times V$ – множество пар вершин графа (u, v), называемых *ребрами*.
- Граф *неориентированный (undirected)*, если вершины ребра не упорядочены, иначе – *ориентированный (directed)*.
- Граф *связный (connected)*, если существует путь из каждой вершины графа в любую другую. Иначе – *несвязный (disconnected)*

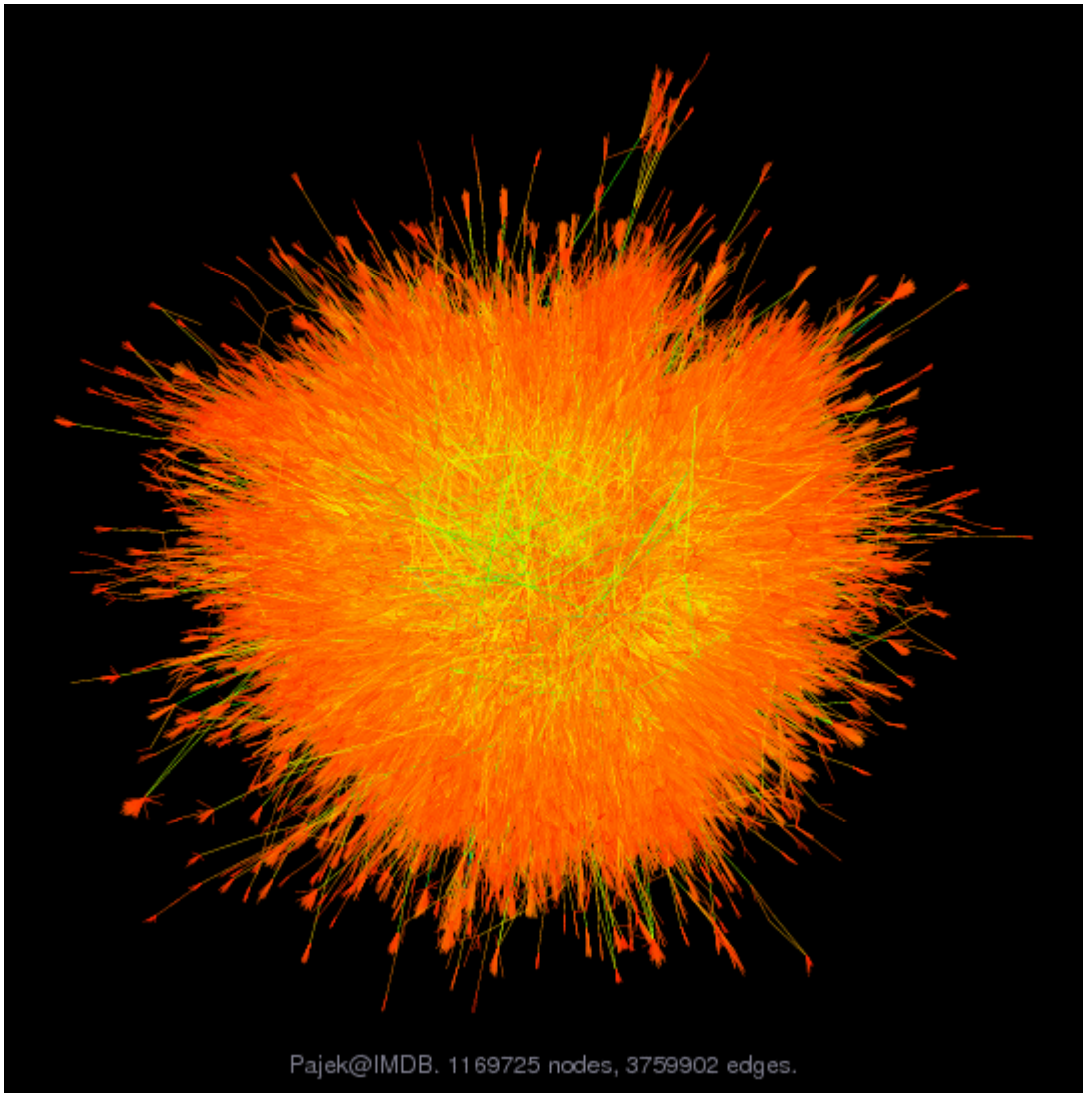
Некоторые типы графов. Двудольный граф

- *Двудольный (bipartite)* граф – граф $G(V, E)$, вершины которого можно разделить на два подмножества V_1 и V_2 так, что любое ребро соединяет вершину одного подмножества только с вершиной другого подмножества
  - Граф описывает отношения между двумя множествами (авторы и публикации / фильмы, поисковые запросы и URL…)



Некоторые типы графов. Двудольный граф

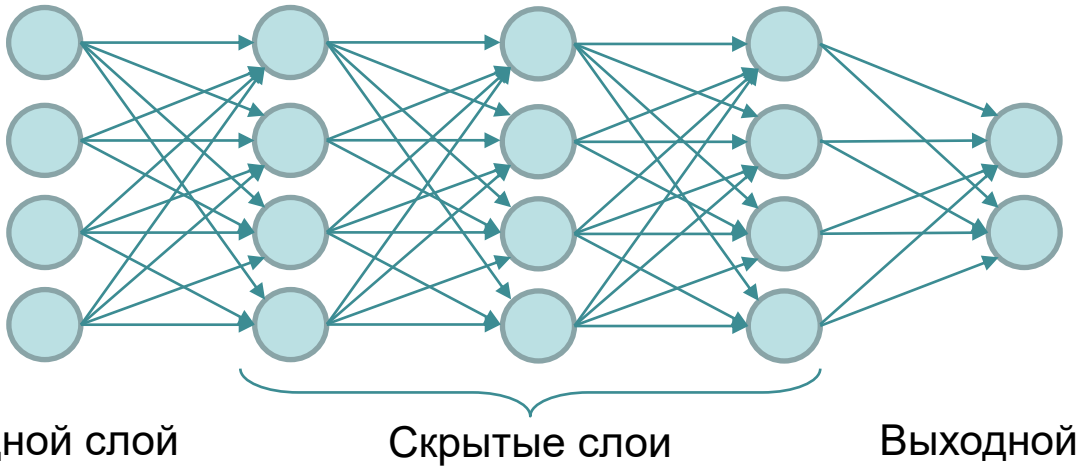
Сеть «актеры - фильмы», основанная на данных сайта www.imdb.com.
A[i, j] = 1, если актер i играл в фильме j.
Граф из коллекции Suite Sparse
https://sparse.tamu.edu/Pajek/IMDB



Некоторые типы графов. DAG

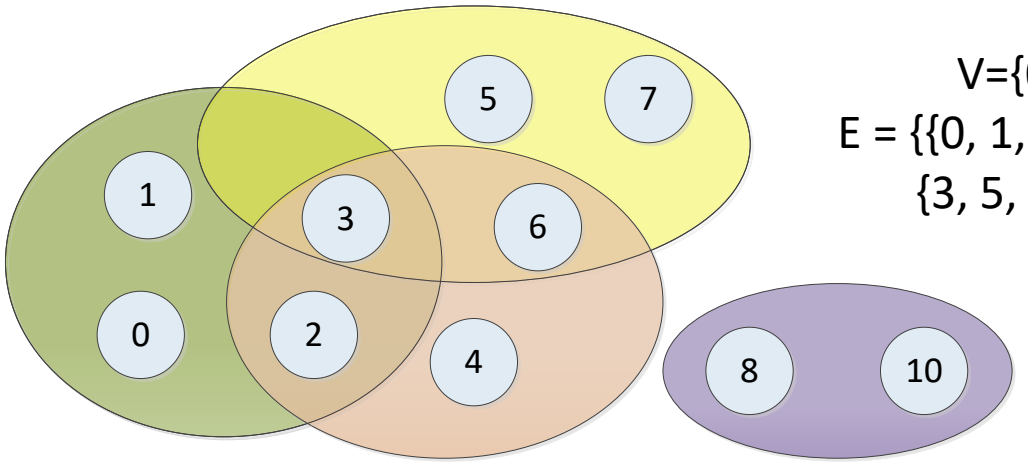
- Ориентированный ациклический граф (DAG, directed acyclic graph) – ориентированный граф, который не содержит циклов. Граф может содержать несколько путей между двумя вершинами.
  - Примеры: история изменений в распределенной системе контроля версий, Байесовские сети, граф цитат и др.
  - Подробнее: https://en.wikipedia.org/wiki/Directed_acyclic_graph

Схема полносвязной нейронной сети (Fully-Connected Neural Network)
Автор: В. Кустикова



Некоторые типы графов. Гиперграф

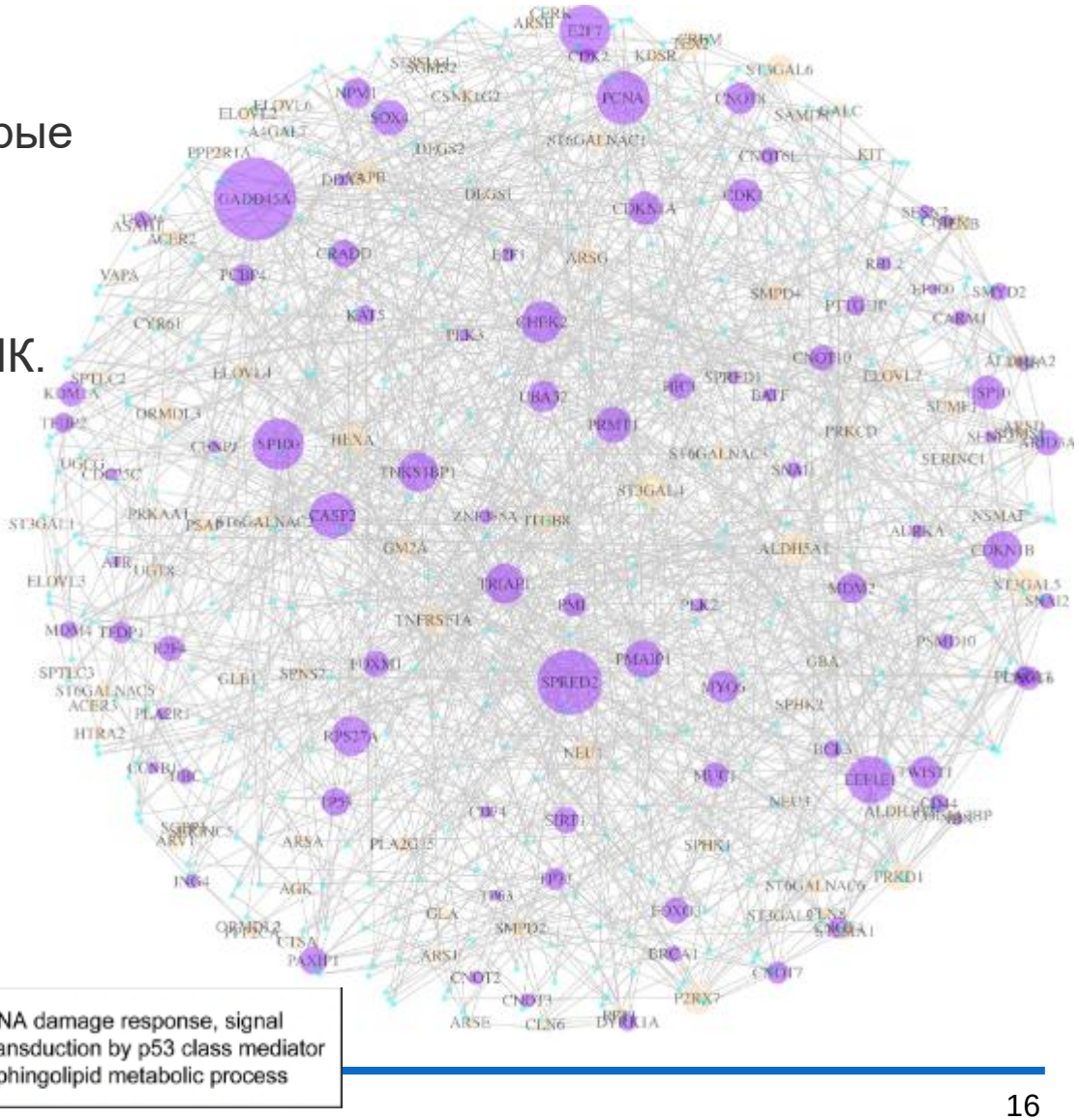
- Гиперграф – обобщение понятия графа. Это граф, в котором ребро может соединять любое число вершин.
  - Примеры: граф интегральной схемы, граф рекомендационной системы, граф биологической системы. Используются в машинном обучении в качестве модели данных и для регуляризации.



V={0, 1, …, 10}
E = {{0, 1, 2, 3}, {2, 3, 4, 6},
{3, 5, 6, 7}, {8, 10}}

Некоторые типы графов. Гиперграф

Задача исследования трехстороннего взаимодействия генов: пара генов, которые изменяют корреляцию, и третьего гена, который отражает основные клеточные состояния.
Визуализация гиперграфа триплетов ДНК.
Размер вершины отражает ее степень.



Kong Y., Yu T. A hypergraph-based method for large-scale dynamic correlation study at the transcriptomic scale //BMC genomics. – 2019. – Т. 20. – №. 1. – С. 397. https://bmcgenomics.biomedcentral.com/articles/10.1186/s12864-019-5787-x/figures/3

ЗАДАЧИ НА ГРАФАХ

Основные задачи на графах

- Обход графа (graph traversal) и поиск в графе – поиск в ширину, поиск в глубину
- Нахождение расстояния между вершинами – кратчайшие пути между двумя вершинами, между всеми вершинами
- Поиск минимального остовного дерева
- Задачи раскраски графа
- Разделение графа, сжатие графа
- Поиск компонент связности, сильных компонент связности
- Задачи для сетей: поиск максимального потока в сети
- Вычисление характеристик сети: PageRank, центральности вершин / ребер, поиск сообществ
- Поиск подграфов, клик, треугольников и др.

Основные задачи на графах

- Особенность большинства задач: задачи NP-полные, на практике применяются эвристические методы
- Нет единой классификации задач
- Классификация из книги Седжвика «Фундаментальные алгоритмы на C++»:

	E	T	I	?
Неориентированные графы				
Связность	•			
Полная связность		•		
Эвклидов цикл	•			
Гамильтонов цикл			•	
Двудольное сочетание	•			
Максимальное сочетание		•		
Планарность		•		
Максимальная клика			•	
Раскраска в 2 цвета	•			
Раскраска в 3 цвета			•	
Кратчайшие пути	•			
Самые длинные пути			•	
Вершинное покрытие			•	
Изоморфизм				•

	E	T	I	?
Орграфы				
Транзитивное замечание	•			
Сильная связность	•			
Цикл нечетной длины	•			
Цикл четной длины		•		
Взвешенные графы				
Минимальное остовое дерево	•			
Задача коммивояжера			•	
Сети				
Кратчайшие пути (неотрицательные веса)	•			
Кратчайшие пути (отрицательные веса)			•	
Максимальный поток	•			
Распределение		•		
Поток минимальной стоимости		•		

Обозначения:

E Легкая – известен эффективный классический алгоритм решения (см. справку)

T Решаемая – решение существует (трудно получить реализацию)

I Нерешаемая – эффективное решение неизвестно (NP-трудная задача)

? Неизвестно, существует ли решение

Типовые приемы

- Основные схемы построения последовательных и параллельных алгоритмов на графах:
  - Жадный принцип
  - Принцип «разделяй и властвуй»
  - Динамическое программирование

Особенности параллельных алгоритмов на графах

- Зависимость вычислений от данных
  - Вычислительная нагрузка неизвестна заранее
  - Алгоритмы часто сформулированы итеративно
- Неструктурированность задач
  - Эффективность параллельного алгоритма зависит от структуры графа. Параллельный алгоритм, подходящий для графов одного вида, может быть не эффективным для графов другого вида.
  - Статическое распределение данных между потоками / процессами может привести к дисбалансу вычислительной нагрузки
  - Отсутствует векторизация

Особенности параллельных алгоритмов на графах

- Низкая локальность
  - Нерегулярный доступ к памяти, кэш-промахи, ошибки предсказания ветвлений
- Низкая арифметическая интенсивность
  - Число операций обращения к памяти значительно превышает число арифметических операций над данными

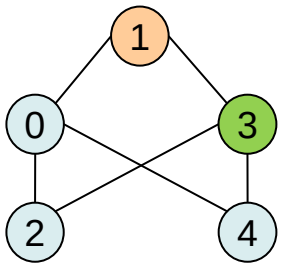
Типовые приемы

- Приемы для построения параллельных алгоритмов:
  - Предобработка графа: разделение по вычислительным узлам, нахождение независимого множества вершин, сжатие структуры, сортировка ребер по весу и др.
  - Отдельная обработка «нетипичных» вершин (например, с большой либо маленькой степенью)
  - Асинхронное выполнение алгоритмов
  - Рандомизация для повышения качества эвристических алгоритмов
  - Гибридизация алгоритмов: разные способы обработки больших и маленьких подграфов

ФОРМАТЫ ХРАНЕНИЯ ГРАФОВ

Форматы хранения графов (1)
Матрица смежности

- Пусть граф $G = (V, E)$ содержит N вершин и M ребер.
- *Матрица смежности* – плотная матрица A размера $N \times N$, в которой элемент $a_{ij} = 1$, если в графе присутствует ребро между вершинами i и j, иначе $a_{ij} = 0$.
- Способ хранения подходит для *плотных* графов, то есть тех, у которых число ребер порядка $\Theta(N^2)$.
- Для ненаправленных графов матрица смежности симметрична
- Память: $4 * N^2$ байт

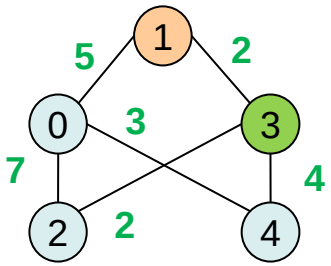


0	1	1	0	1
1	0	0	1	0
1	0	0	1	0
0	1	1	0	1
1	0	0	1	0

Форматы хранения графов (1)
Матрица смежности

- Хранение информации о весах:

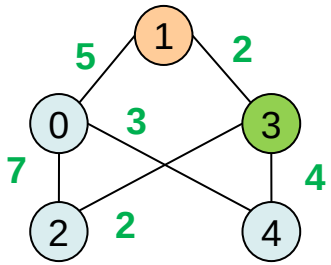
$$a_{ij} = \begin{cases} w(i,j), i \text{ и } j \text{ связаны ребром} \\ 0, i = j \\ \infty, i \text{ и } j \text{ не связаны} \end{cases}$$



0	5	7	∞	3
5	0	∞	2	∞
7	∞	0	2	∞
∞	2	3	0	4
3	∞	∞	4	0

Форматы хранения графов (2). Массив ребер

- Массив ребер **Edges** – ребра графа хранятся в произвольном порядке, для каждого ребра хранятся последовательно инцидентные ему вершины. Размер массива $2 * M$.
- Хранение информации о весах ребер:
  - **Eweights** – массив весов ребер размера M. **Eweights[i]** хранит вес ребра **Edges[i / 2]**
- Формат удобен для некоторых алгоритмов (например, поиск компонент связности или остовного дерева)



Edges

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

Eweights

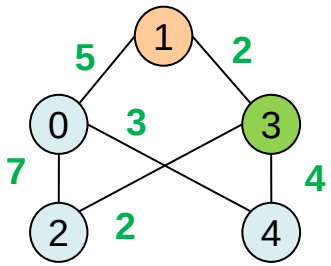
5	7	3	2	2	4

Форматы хранения графов (3). Списки смежности, CRS

- Списки смежности (формат хранения матрицы Compressed Row Storage, CRS)
- Пусть граф $G = (V, E)$ содержит N вершин и M ребер. Хранение в двух массивах:
  - **Adjncy** – хранит номера вершин, связанных с данной, последовательно для вершин 0, 1, …, $N-1$. Размер массива $2 * M$.
  - **Xadj –** хранит индексы начала списка смежности каждой вершины в массиве **Adjncy**. Дополнительно хранится «фиктивный» элемент – индекс последнего элемента массива **Adjncy,** увеличенный на единицу. Размер массива $N + 1$.
- Вершины, смежные с вершиной i, хранятся в массиве **Adjncy** по индексам от **Xadj[i]** до **Xadj[i + 1] – 1** включительно.

Форматы хранения графов (3). Списки смежности, CRS

- Способ хранения подходит для *разреженных* графов, то есть тех, у которых число ребер одного порядка с числом вершин.
- Память: $4 * (2 * M + N + 1) = 8 * M + 4 * N + 4$ байт
- Хранение информации о весах ребер:
  - **Eweights** – массив весов ребер размера $2 * M$. **Eweights[i]** хранит вес ребра **Adjncy[i]**



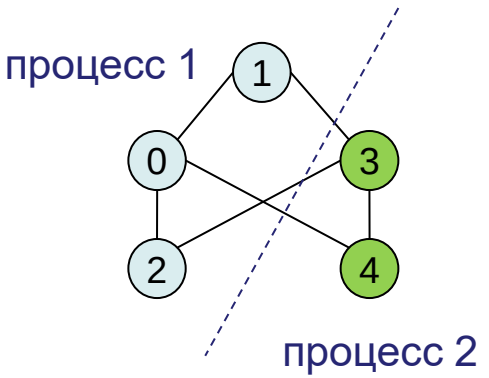
Adjncy	1	2	4	0	3	0	3	1	2	4	0	3

Xadj	0	3	5	7	10	12

Eweights	5	7	3	5	2	7	2	2	3	4	3	4

Форматы хранения графов (4). Распределенные списки смежности (CRS)

- Хранение графа на P процессах в CRS формате:
  - массивы **Adjncy**, **Xadj** для локальных вершин. Нумерация с 0.
  - **VertexDist** – массив размера $P + 1$, описывает распределение вершин между процессами. Для процесса i локальными являются вершины с **VertexDist[i]** по **VertexDist[i+1]–1** включительно. Хранится на каждом процессе



процесс 1

Adjncy	1	2	4	0	3	0	3
Xadj	0	3	5	7			
VertexDist	0	3	5				

процесс 2

Adjncy	1	2	4	0	3
Xadj	0	3	5		
VertexDist	0	3	5		

Заключение

- Представление данных в виде графа используется во многих прикладных областях: биологические, транспортные, коммуникационные сети, энергетические системы, веб-графы, графы социальных сетей и др.
- Наряду с графами общего вида, ряд моделей описывается графами специального вида. Как правило, для таких графов создаются отдельные алгоритмы обработки.
- Типичные задачи обработки графов – обход графа, поиск кратчайших путей, разделение, вычисление характеристик графа и др. Ряд задач имеет точное решение, другие NP-трудные, решаются эвристическими методами.

Заключение

- Для параллельных алгоритмов обработки графов характерно отсутствие единообразной структуры данных, зависимость вычислений от данных, низкая арифметическая интенсивность. Данные в виде графа плохо «ложатся» на архитектуру памяти устройства.
- Для повышения эффективности параллельных алгоритмов на графах применяется ряд приемов: предобработка данных, асинхронизация вычислений, комбинирование алгоритмов.
- Хранение графов: для плотных графов используется представление в виде матрицы смежности, для разреженных графов – списки смежности (аналог CRS). При вычислениях на распределенной памяти используются распределенные списки смежности.

Общая литература по курсу

1. Т. Кормен, Ч. Лейзерсон, Р. Ривест, К. Штайн. Алгоритмы: построение и анализ, 3-е издание. – М.: «Вильямс», 2013. – 1328 с.
2. Седжвик Р. Фундаментальные алгоритмы на C++. Алгоритмы на графах: Пер. с англ./Роберт Седжвик //СПб: ООО «ДиаСофтЮП». – 2002.
3. Grama A. et al. Introduction to parallel computing. – Pearson Education, 2003. https://www-users.cs.umn.edu/~karypis/parbook/
4. Erciyes K. Guide to Graph Algorithms. – Springer International Publishing, 2018.
5. Newman M. Networks. – Oxford university press, 2018.
6. Easley D. et al. Networks, crowds, and markets. – Cambridge: Cambridge university press, 2010. – Т. 8. https://www.cs.cornell.edu/home/kleinber/networks-book/
7. Список литературы к курсу проф. Шана – ссылки на научную литературу по разным задачам теории графов https://people.csail.mit.edu/jshun/graph.shtml

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