БИБЛИОТЕКА NVGRAPH: ПРИМИТИВЫ ДЛЯ РАБОТЫ С ГРАФАМИ

Марат Арсаев

NVIDIA

01.03.2018

СОДЕРЖАНИЕ

- Обзор концептов, форматов данных и возможностей nvGRAPH
- Новое в CUDA 9.0:
  - Обход графа в ширину
  - Кластеризация графа
  - Стягивание графа
- Дальнейшая работа
- Q&A

ОСНОВЫ NVGRAPH

ОСНОВЫ NVGRAPH

- Статичные графы
- Single GPU
- Форматы структуры – Edge List, CSR, CSC
- Конвертация форматов
- Несколько наборов весов для вершин и ребер
- Менеджер памяти [1]

[1] https://github.com/NVIDIA/cnmem

ПОДДЕРЖИВАЕМЫЕ ПРИМИТИВЫ

- Матрица×Вектор[1] с операциями полукольца:
  - (+, ×)
  - (min, +)
  - (max, min)
  - (OR, AND)
- Widest Path
- Single Source Shortest Path
- Pagerank
- Вычисление подграфа
- Подсчет треугольников [2]

[1] Merrill, D. and Garland, M. 2015. Merge-based Parallel Sparse Matrix-Vector Multiplication using the CSR Storage Format. Tech. Rep. NVR-2015-002, NVIDIA Corp.

[2] M. Bisson and M. Fatica, "High Performance Exact Triangle Counting on GPUs," in *IEEE Transactions on Parallel and Distributed Systems*, vol. 28, no. 12, pp. 3501-3510, Dec. 1 2017.

ОБХОД ГРАФА В ШИРИНУ

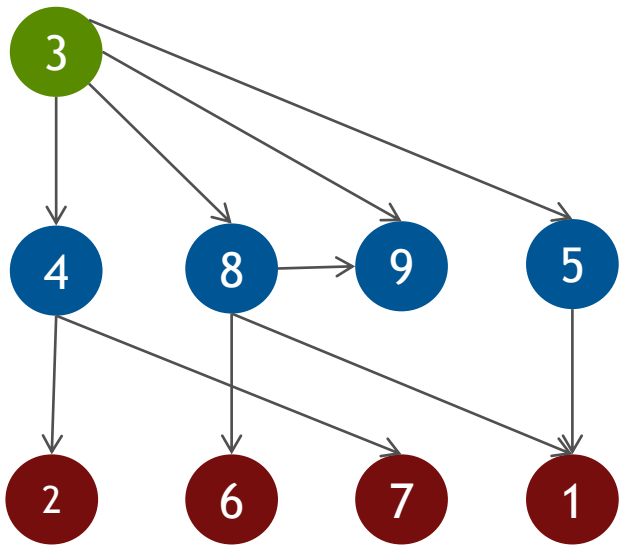
ОБХОД ГРАФА

Шаг сверху-вниз

Начальный фронт

Фронт, глубина = 1

Фронт, глубина = 2

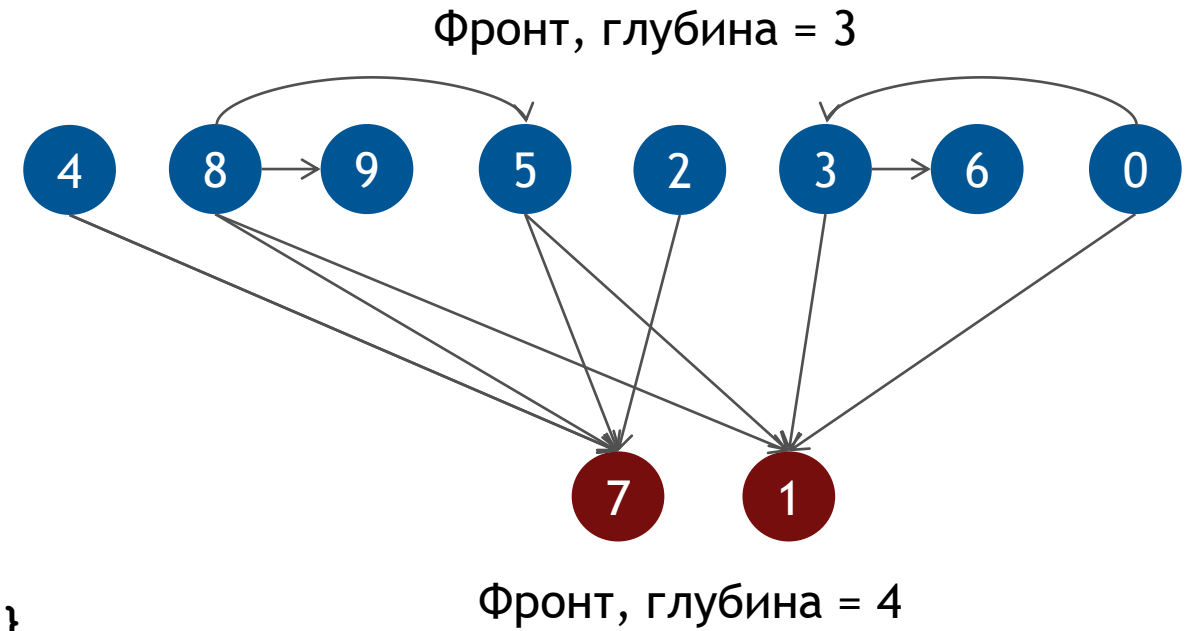



```
for v ∈ frontier
    for n ∈ neighbors[v]
       if parents[n] = -1
           parents[n] ← v
           next ← next U {n}
```

ОБХОД ГРАФА

Шаг снизу-вверх

- Иногда лучше искать кандидатов для следующего шага наоборот




```
for v ∈ vertices
  if parents[v] = -1
    for n ∈ neighbors[v]
      if n ∈ frontier
        parents[v] ← n
        next ← next U {v}
        break
```

ОБХОД ГРАФА

Результаты

NVIDIA K40

Graph	GTEPS - Gunrock	GTEPS - nvGraph
rmat_n22_e64	119.1	109.3
rmat_n23_e32	62.1	65.8
rmat_n24_e16	30.6	38.31
hollywood-2009	19.1	18.7
roadNet-CA	0.32	0.12

NVIDIA P100

Graph	GTEPS - Gunrock	GTEPS - nvGraph
rmat_n22_e64	291	319
rmat_n23_e32	165	-
rmat_n24_e16	88	128
roadNet-CA	-	0.14

CPU GTEPS is <= 2

ОБХОД ГРАФА

Ссылки

- D. Merrill, M. Garland, and A. Grimshaw, Scalable GPU graph traversal
- Mauro Bisson, Massimo Bernaschi, and Enrico Mastrostefano, Parallel Distributed Breadth First Search on the Kepler Architecture
- Scott Beamer, Krste Asanovic and David Patterson, Direction-Optimizing Breadth-First Search, Direction-Optimizing Breadth-First Search
- Yangzihao Wang, Andrew Davidson, Yuechao Pan, Yuduo Wu, Andy Riffel, John D. Owens, Gunrock : A High-Performance Graph Processing Library on the GPU
- Gunrock benchmarks

КЛАСТЕРИЗАЦИЯ

ПОСТАНОВКА ЗАДАЧИ

Сгруппировать ‘близкие’ вершины вместе

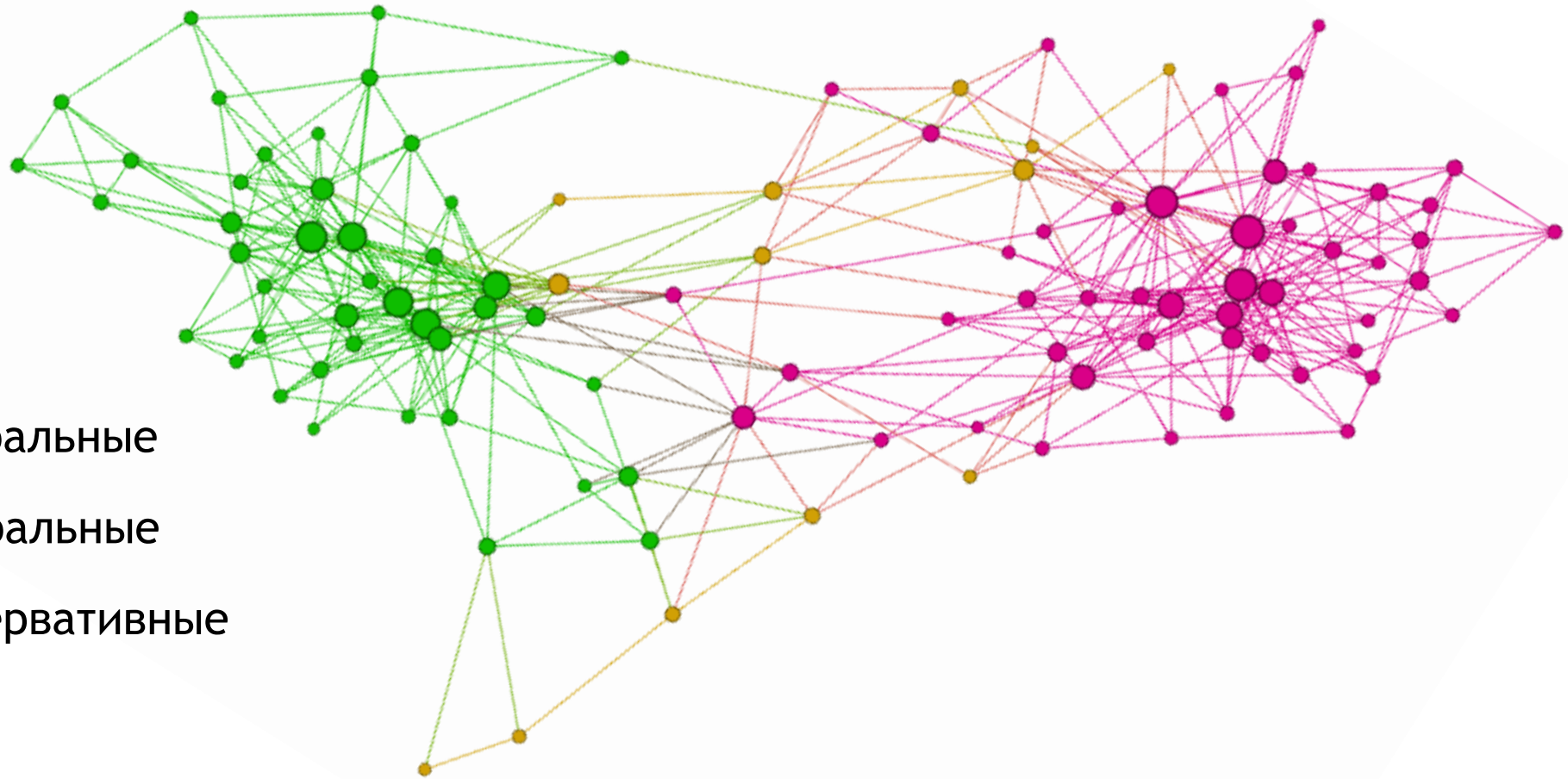
Например: определить группы, базируясь на данных о покупках тематических книг с Amazon.com

Data: V. Krebs. 2004

Visualization : M. Bastian, S. Heymann, and M. Jacomy. 2009. “Gephi: An Open Source Software for exploring and manipulating networks”

ПОСТАНОВКА ЗАДАЧИ

Контрольные кластеры



Розовый Либеральные

Желтый Нейтральные

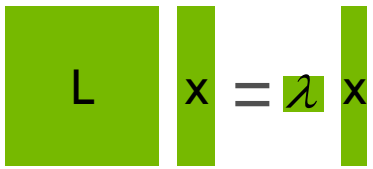
Зеленый Консервативные

Data: V. Krebs. 2004

Visualization : M. Bastian, S. Heymann, and M. Jacomy. 2009. “Gephi: An Open Source Software for exploring and manipulating networks”

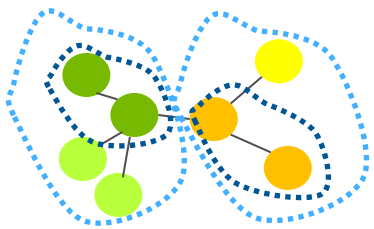
АЛГОРИТМЫ КЛАСТЕРИЗАЦИИ

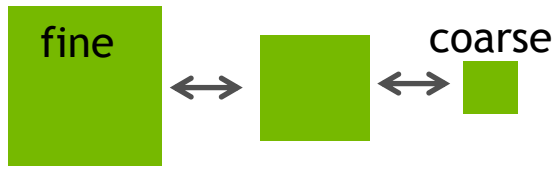
- Спектральные



Выразить задачу в матричном виде, решить задачу собственных значений и использовать собственные вектора

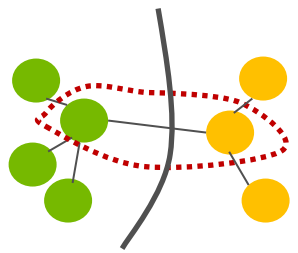
- Иерархические





Построить иерархию графов, разделить уменьшенный граф, распространить разделение на верхние уровни

- Местные улучшения

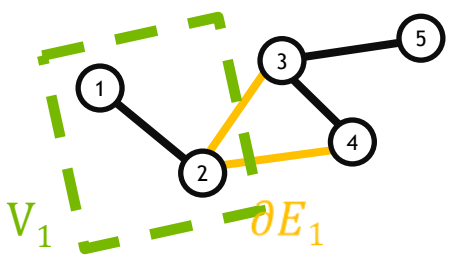


Перекрашивать определенные вершины в уже существующем разделении

БАЛАНСИРОВАННЫЙ РАЗРЕЗ

Задача минимизации стоимости разреза

$$\min_{V_i} \sum_{i=1}^{p} \frac{|\partial E_i|}{|V_i|}$$



$$\min_{x_i} \sum_{i=1}^{p} \frac{x_i^T L x_i}{x_i^T x_i}$$

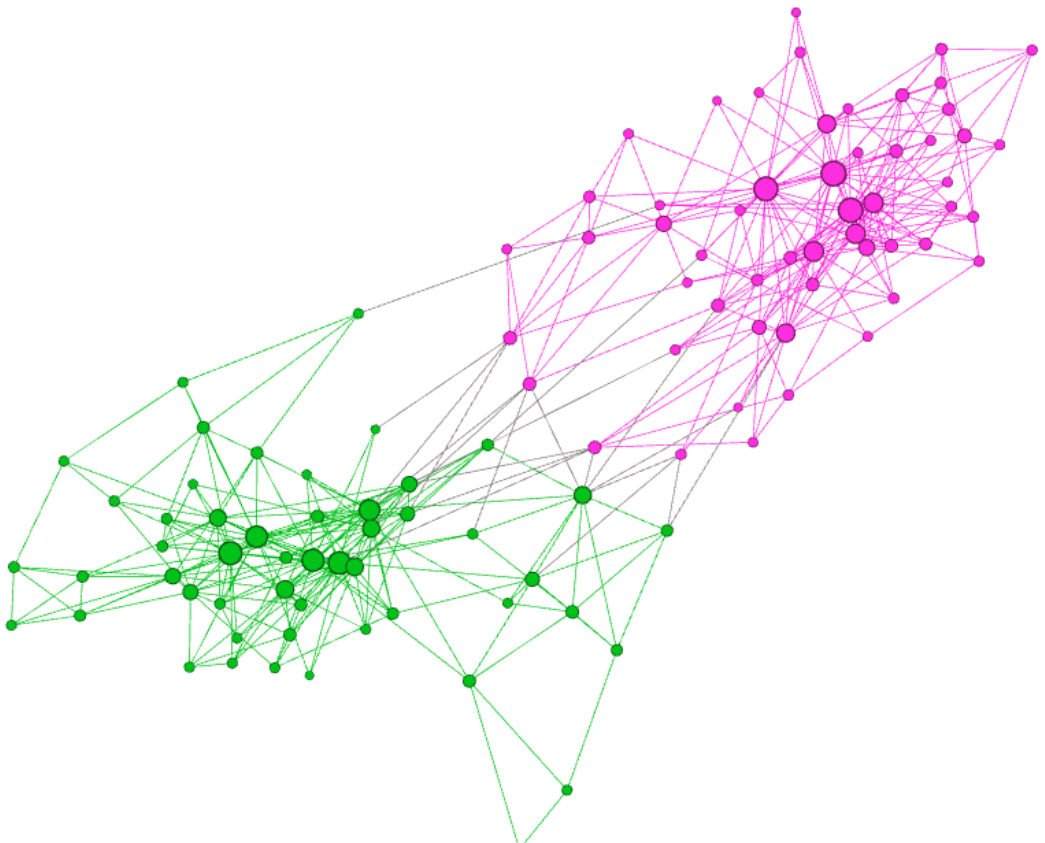
where $x_i \in \{0,1\}^n$ and $x_i \perp x_j$

L = D - A
D - матрица степеней вершин
A - матрица смежности
L - лапласиан графа

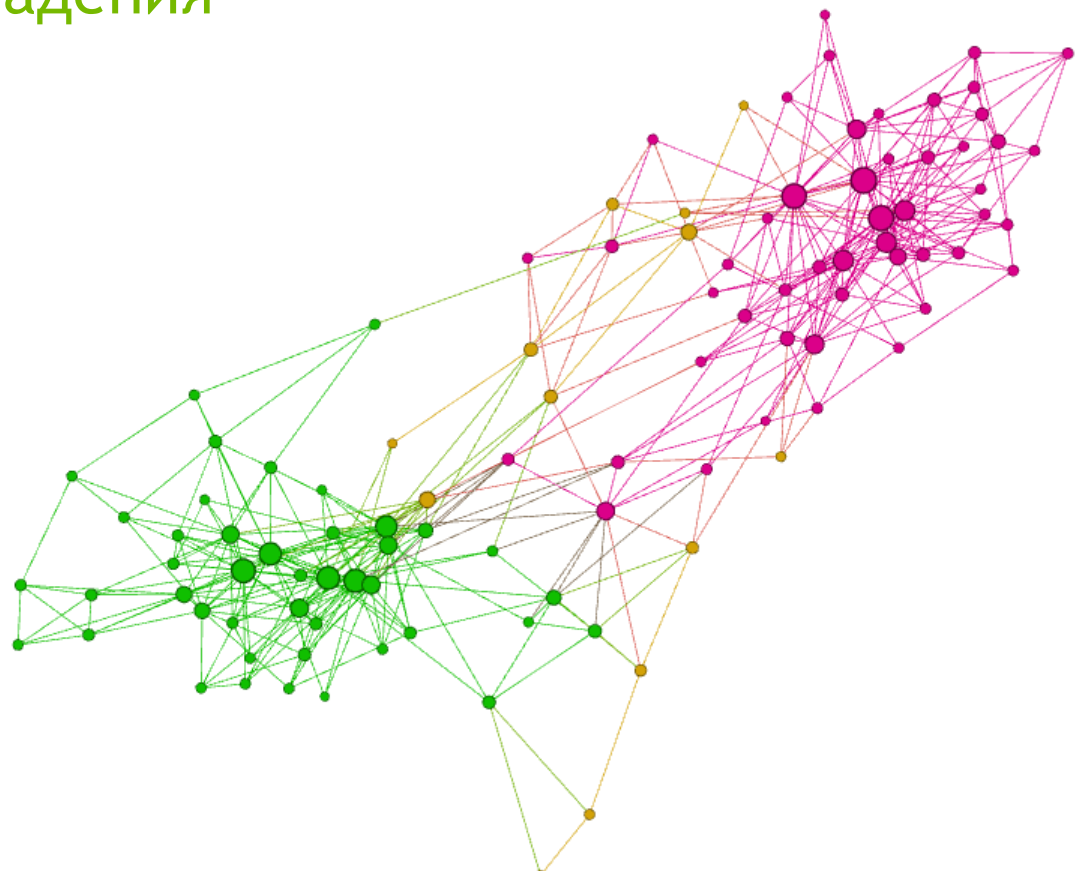
https://research.nvidia.com/publication/parallel-spectral-graph-partitioning

БАЛАНСИРОВАННЫЙ РАЗРЕЗ

80% совпадения



Балансированный разрез

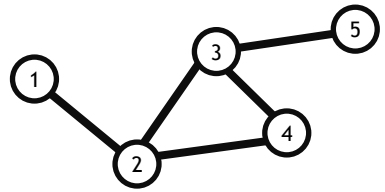


Контрольные кластеры

МАКСИМИЗАЦИЯ МОДУЛЯРНОСТИ

Модулярность – функция разницы между текущим разбиением и случайным разбиением

Для неявной матрицы $$B = A - \frac{1}{2\omega} vv^T$$



Модулярность: $$Q = \frac{1}{2\omega} Tr(X^T BX)$$

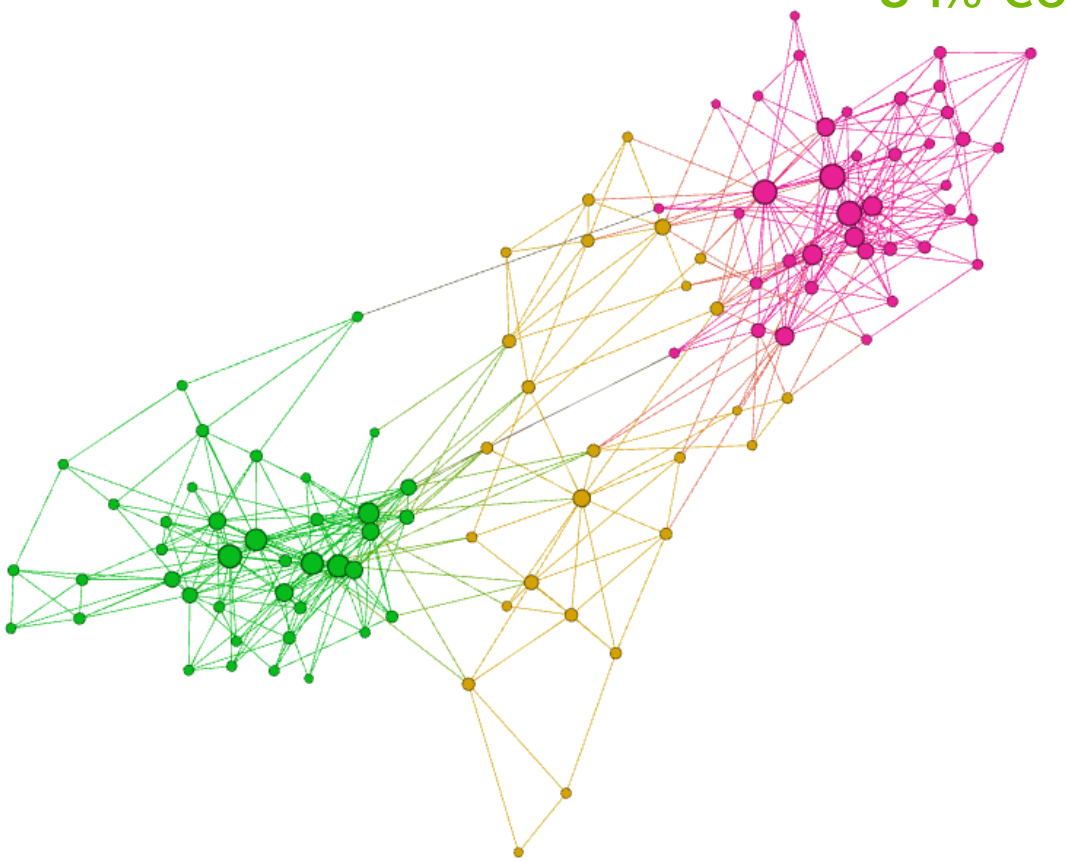
Где $Tr(.)$ - след матрицы, $X = [x_1, \ldots, x_n]$, такие что $x_{ik} = 1$ если $c(i) = k$

$$A = \begin{bmatrix} & 1 & & & \\ 1 & & 1 & 1 & \\ & 1 & & 1 & 1 \\ & 1 & 1 & & \\ & & 1 & & \end{bmatrix} \qquad v = \begin{bmatrix} 1 \\ 3 \\ 3 \\ 2 \\ 1 \end{bmatrix}$$

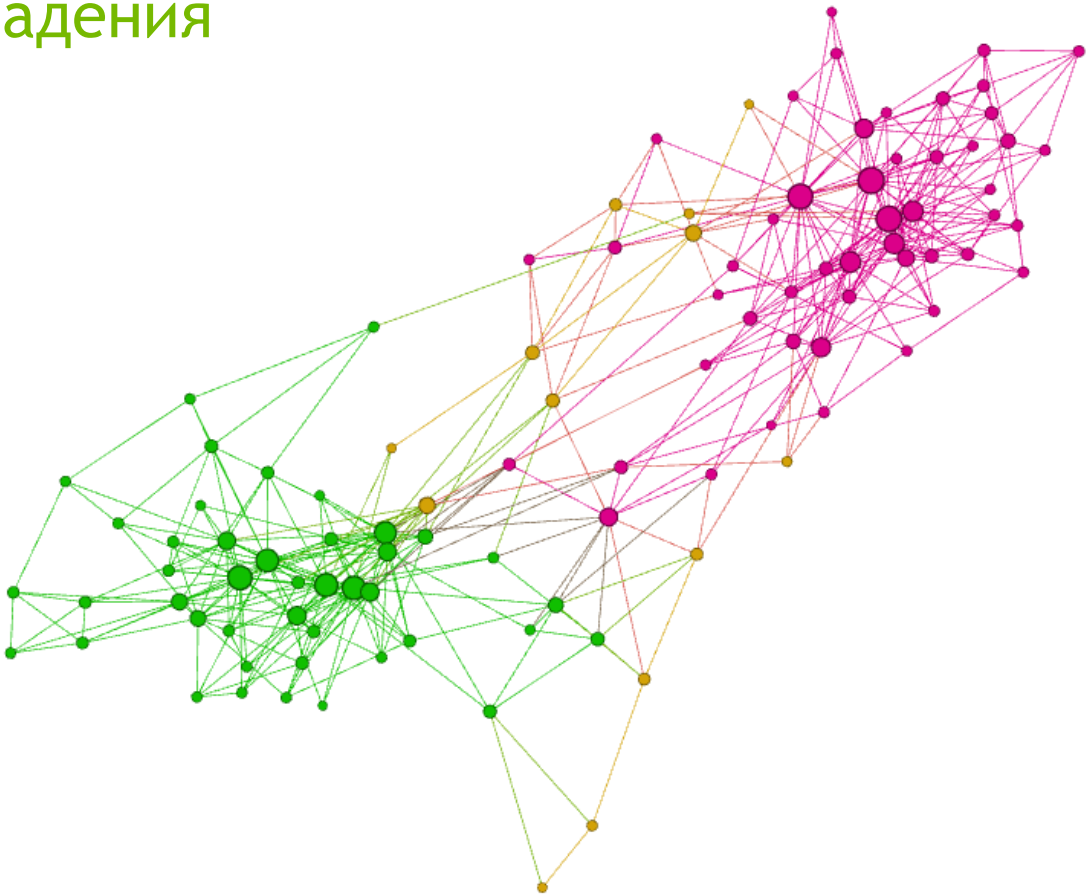
A. Fender, N. Emad, S. Petiton, M. Naumov. 2017. “Parallel Modularity Clustering.” ICCS

МАКСИМИЗАЦИЯ МОДУЛЯРНОСТИ

84% совпадения



Максимизация модулярности

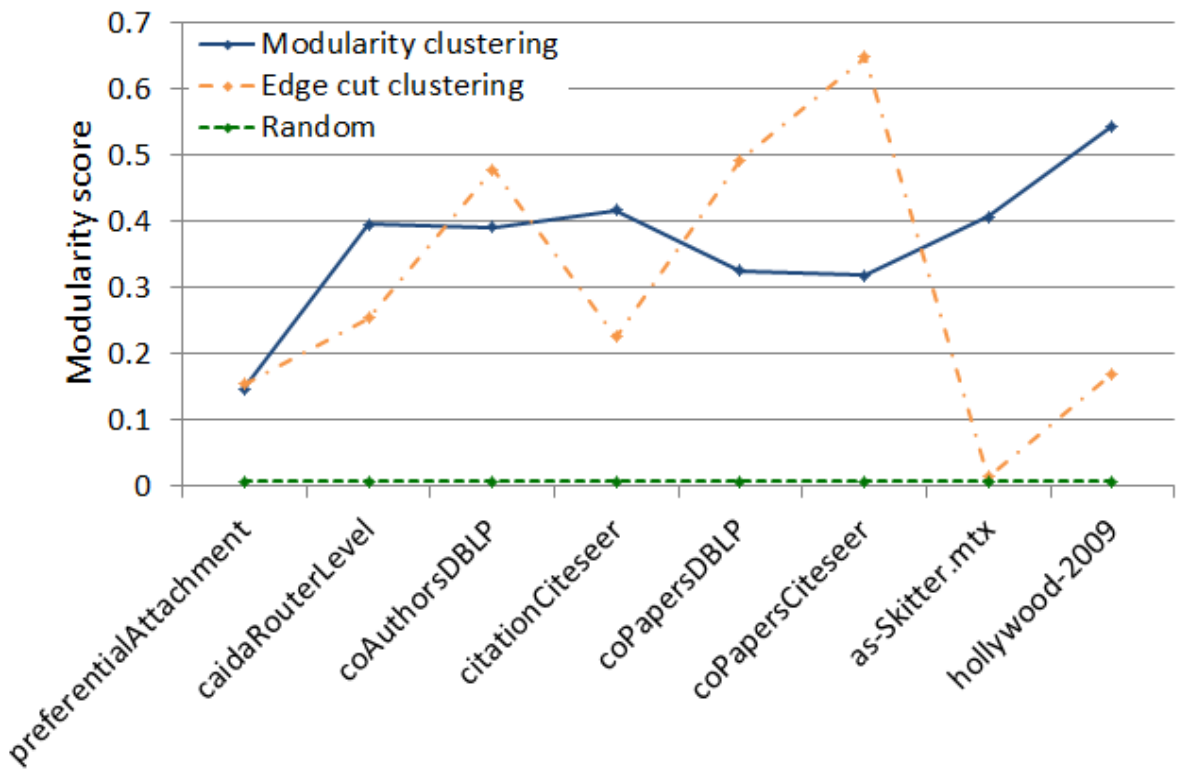


Контрольные кластеры

A. Fender, N. Emad, S. Petiton, M. Naumov. 2017. “Parallel Modularity Clustering.” ICCS

СРАВНЕНИЕ МОДУЛЯРНОСТИ И ЛАПЛАСИАНА

В среднем быстрее методов лапласиана в 3 раза



#	Matrix	$n = \|V\|$	$m = \|E\|$	Application
1.	preferentialA...	100,000	499,985	Artificial
2.	caidaRouterLevel	192,244	609,066	Internet
3.	coAuthorsDBLP	299,067	977,676	Coauthorship
4.	citationCiteseer	268,495	1,156,647	Citation
5.	coPapersDBLP	540,486	15,245,729	Affiliation
6.	coPapersCiteseer	434,102	16,036,720	Affiliation
7.	as-Skitter	1,696,415	22,190,596	Internet
8.	hollywood-2009	1,139,905	113,891,327	Coauthorship

Модулярность:
более высокие и стабильные показатели модулярности

Лапласиан:
более сбалансированные кластеры

Nvidia Titan X (Pascal)
Intel Core i7-3930K @3.2 GHz

A. Fender, N. Emad, S. Petiton, M. Naumov. 2017. “Parallel Modularity Clustering.” ICCS

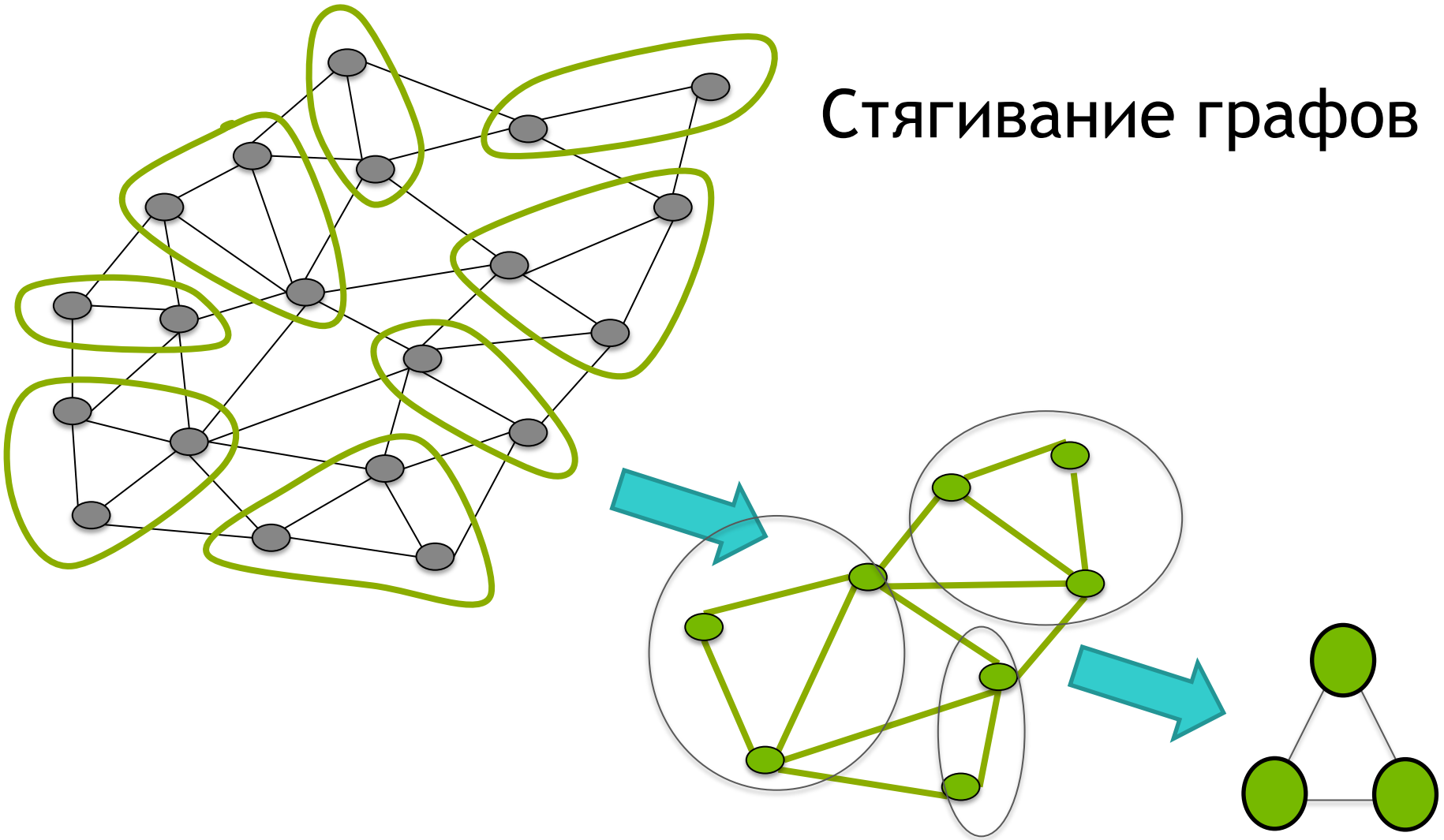
СПЕКТРАЛЬНАЯ КЛАСТЕРИЗАЦИЯ

- Общий интерфейс
- Минимальный сбалансированный разрез [1]
  - Балансирование учтено в функции стоимости
  - Возможно использование различного решателя собственных значений
- Максимизация модулярности [2]
  - Широко используется в анализе социальных сетей
  - Быстрее вычисляется

[1] https://research.nvidia.com/publication/parallel-spectral-graph-partitioning
[2] A. Fender, N. Emad, S. Petiton, M. Naumov. 2017. “Parallel Modularity Clustering.” ICCS

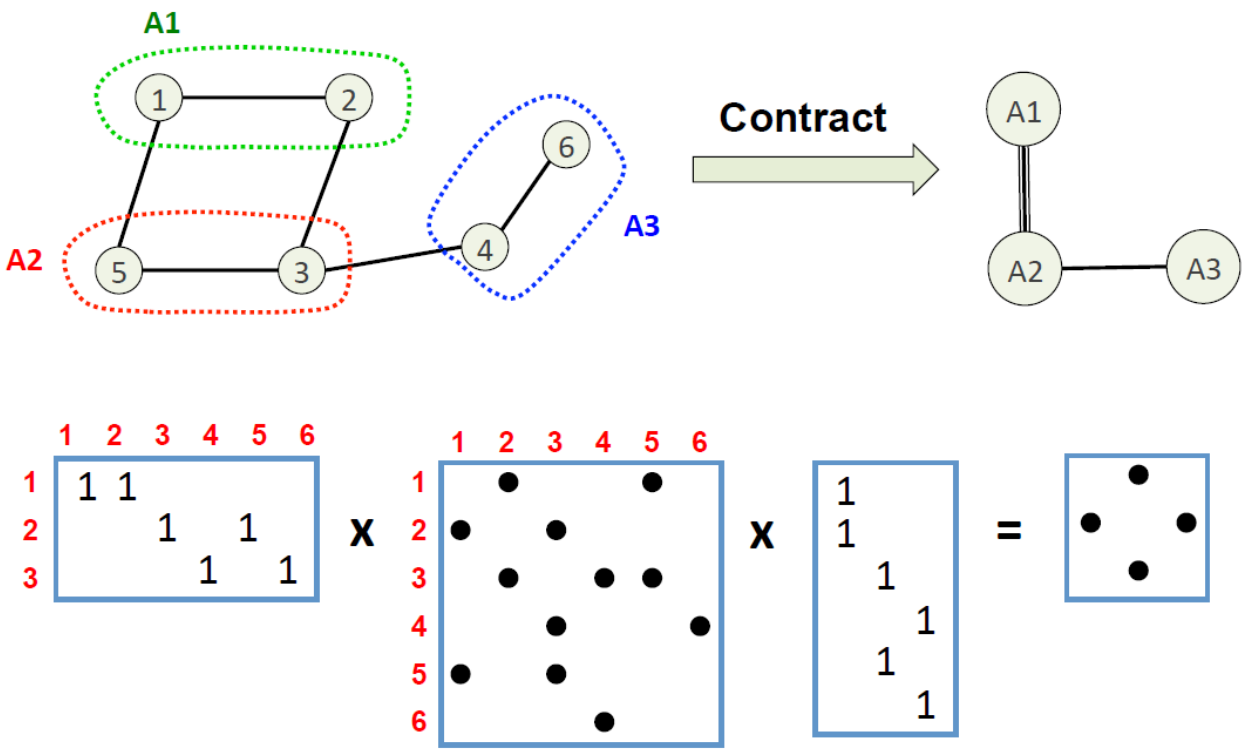
СТЯГИВАНИЕ ГРАФОВ

Стягивание графов



СТЯГИВАНИЕ ГРАФОВ

Матричная постановка: $S*A*S^T$



http://docs.nvidia.com/cuda/cusparse/#cusparse-lt-t-gt-csrgemm
https://github.com/NVIDIA/AMGX/blob/master/core/src/aggregation/aggregation_amg_level.cu

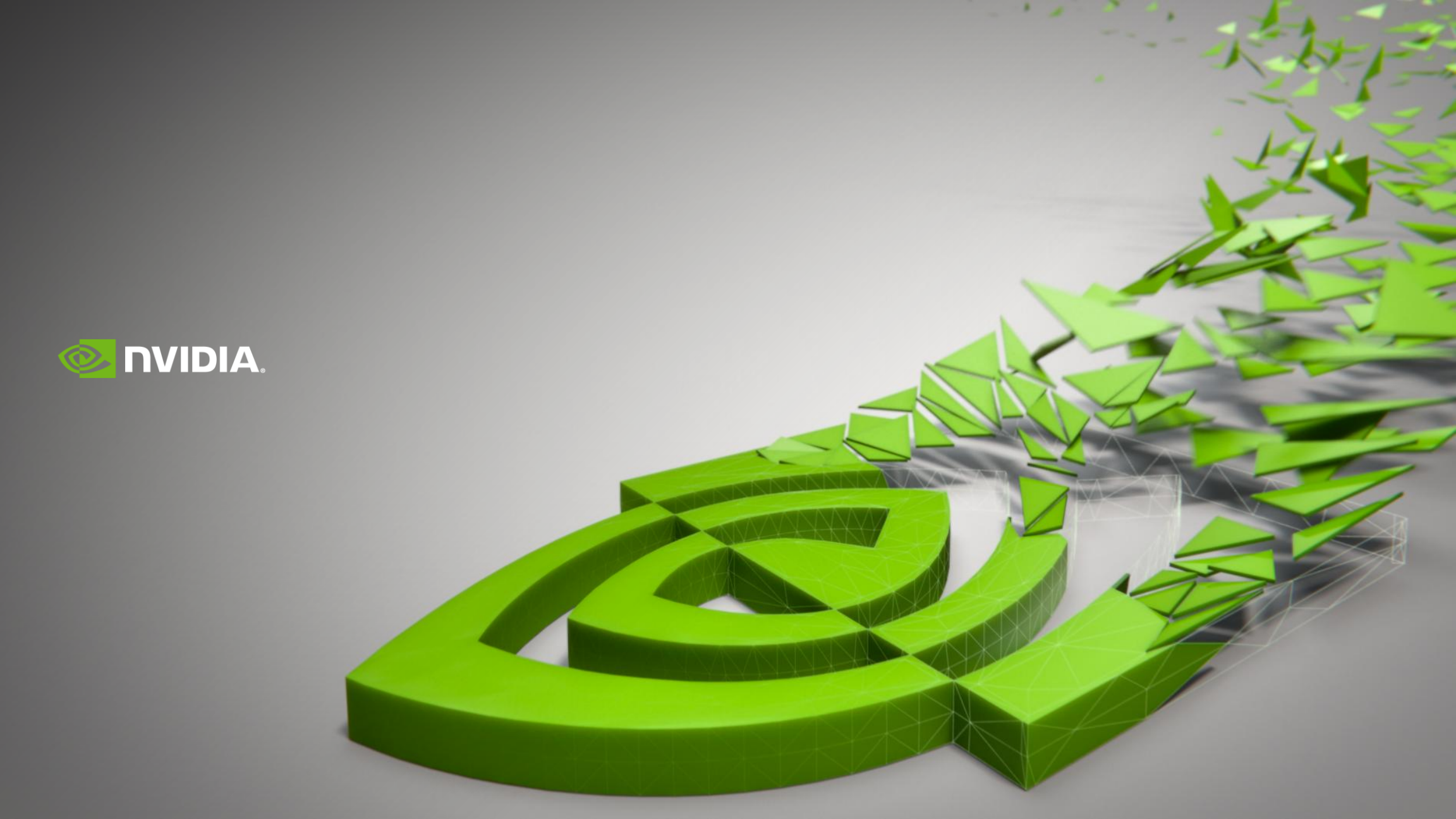
ДАЛЬНЕЙШАЯ РАБОТА

ДАЛЬНЕЙШАЯ РАБОТА

- MultiGPU
- Динамические графы
- Поддержка примитивов и метрик
  - betweenness centrality
  - раскраска и перекраска графов [1]
- Графовые базы данных
- Интеграция с системами аналитики и машинного обучения

[1] https://github.com/NVIDIA/AMGX/tree/master/core/src/matrix_coloring

ВОПРОСЫ?


NVIDIA