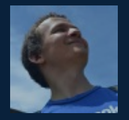
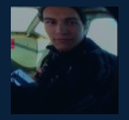
Darwini: Геренация реалистичных синтетических графов



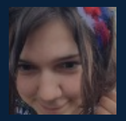
Sergey Edunov
Facebook



Dionysios Logothetis
Facebook

Cheng Wang
University of Houston

Avery Ching
Facebook



Maja Kabiljo
Facebook

Проблема

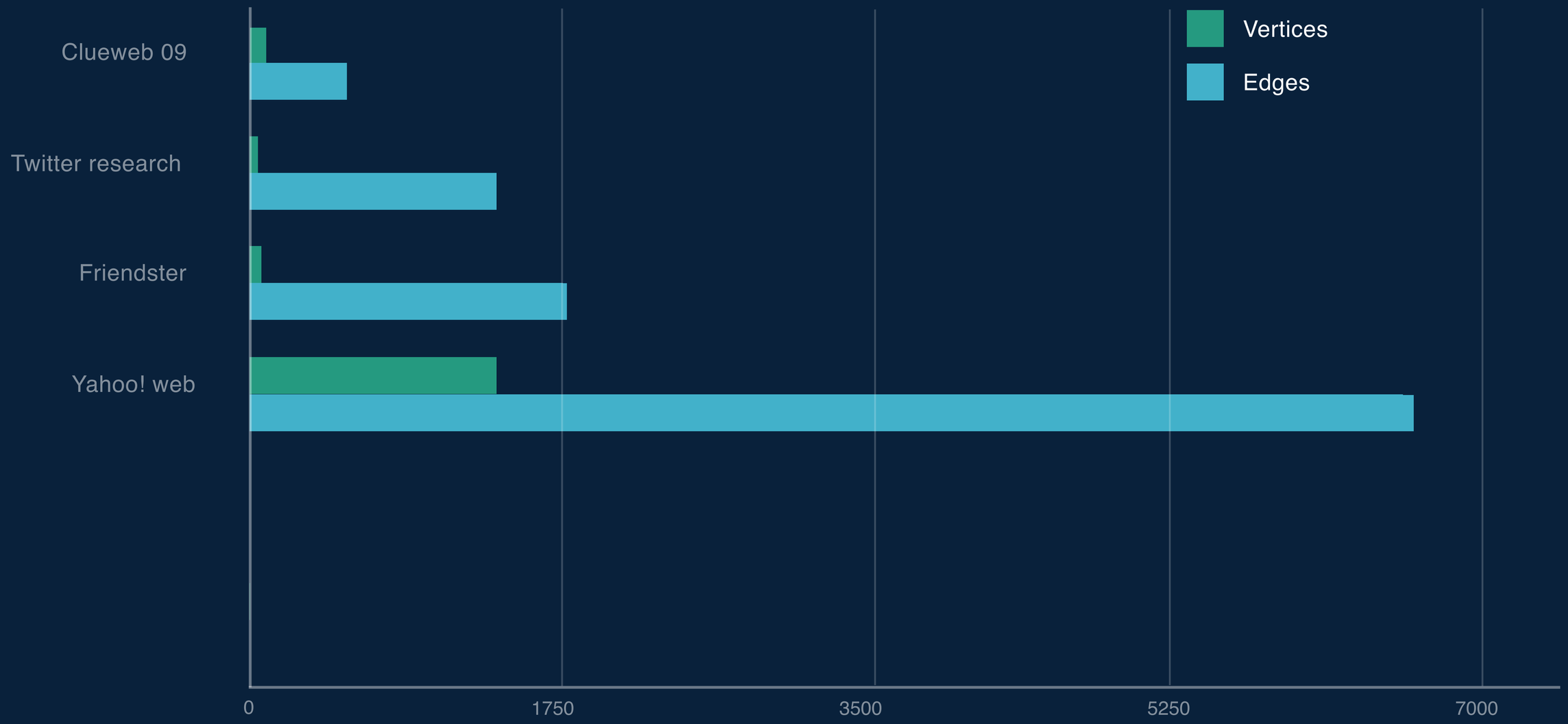
Нам самим нужны большие графы для тестирования проекций

Проблема:

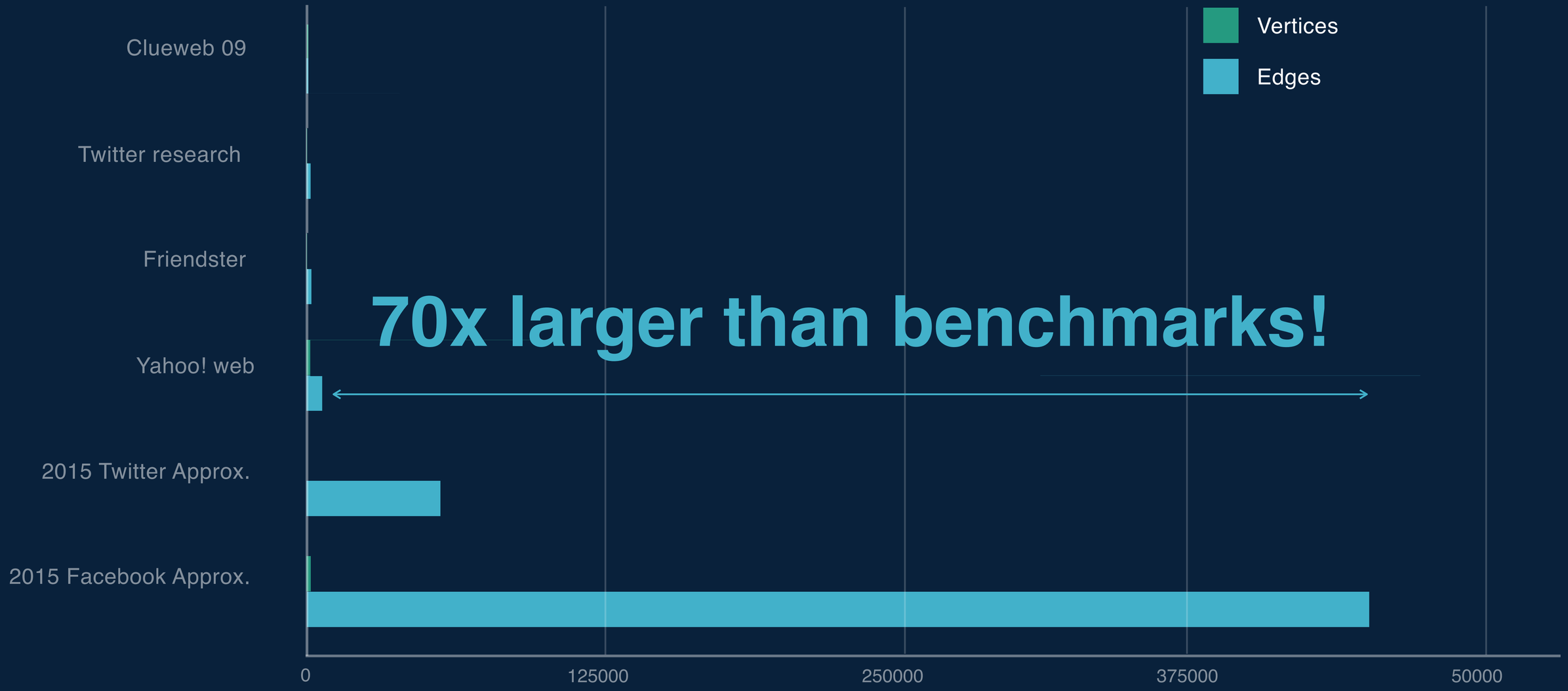
Отсутствие стандартных тестов, которые бы были доступны широкому кругу разработчиков и интересны в индустрии

1. Большинство публикуемых статей сравнивают результаты с Apache Giraph
2. Все публикуемые статьи демонстрируют успешные улучшения

Benchmark Graphs



Benchmark to Social Graphs



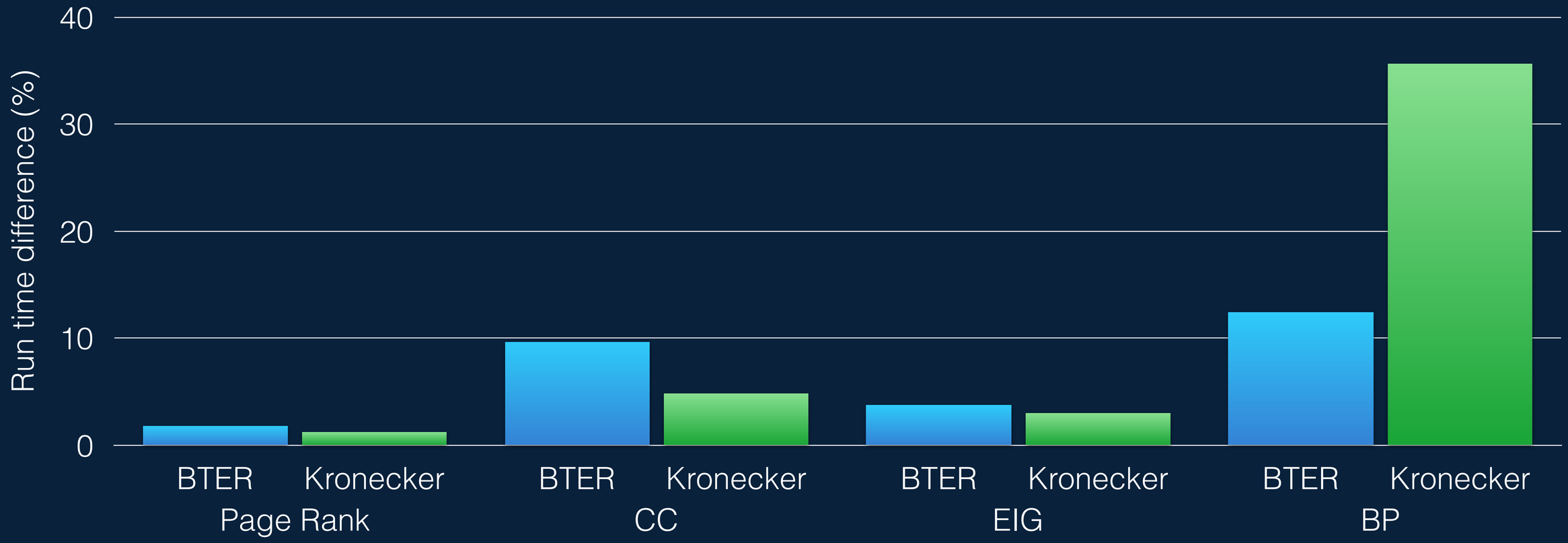
Существующие benchmarks

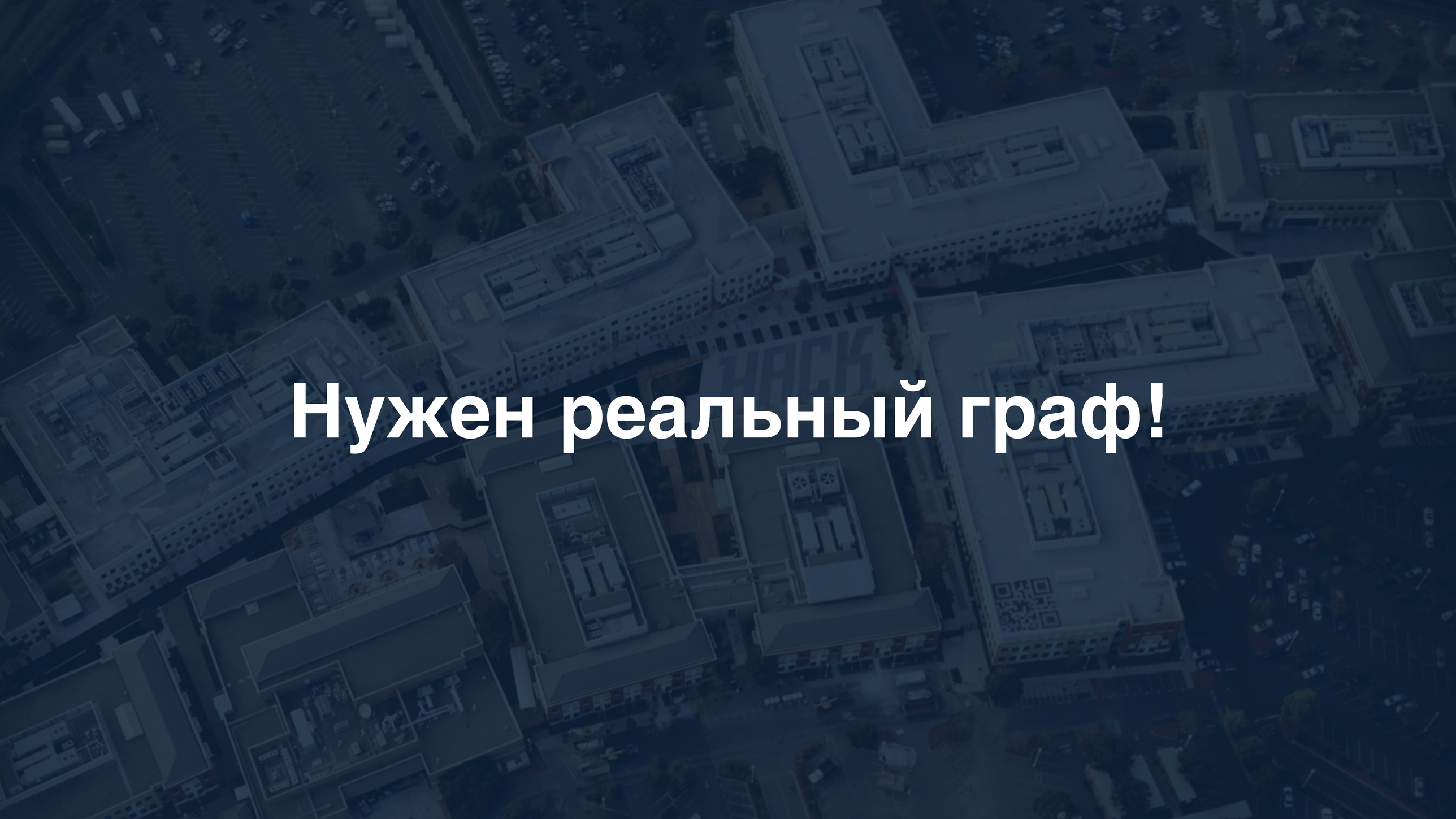
graph500.org

- Поиск в ширину (BFS)
- Kronecker graph

Не интересен в FB

Важность использования реальных графов




Нужен реальный граф!

Может быть синтетический граф?

Erdos Renyi

BTER

Kronecker

R-MAT

LDBC

Random Walk

DK-2

Критерии поиска

1. Масштабируемость. Если алгоритм не может создать граф с триллионом ребер, он нам не подходит.
2. Способность передать распределение степеней вершин
3. Способность передать более высоки распределения в терминах DK-серий
4. Способность передать высокоуровневые метрики

Existing algorithms vs requirements

	Kronecker	BTER	Erdos-Renyi
Scalability			
Degree distribution			
Joint degree & CC			
High level metrics			

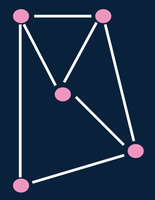
Darwini*

1. Работает на Apache Giraph, выполняется параллельно на сотнях машин
2. Способен производить графы с триллионами ребер
3. Способен производить графы с заданным распределением степеней ребер и коэффициентом кластеризации
4. Лучше передает высокоуровневые метрики

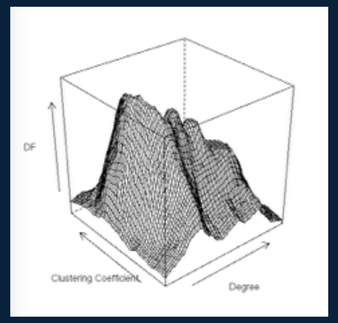
*Caerostris darwini - паук известный самыми большими в мире паутинами, размеры паутин от 900 до 28000 см^2

Использование Darwin

Original Graph

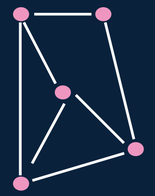


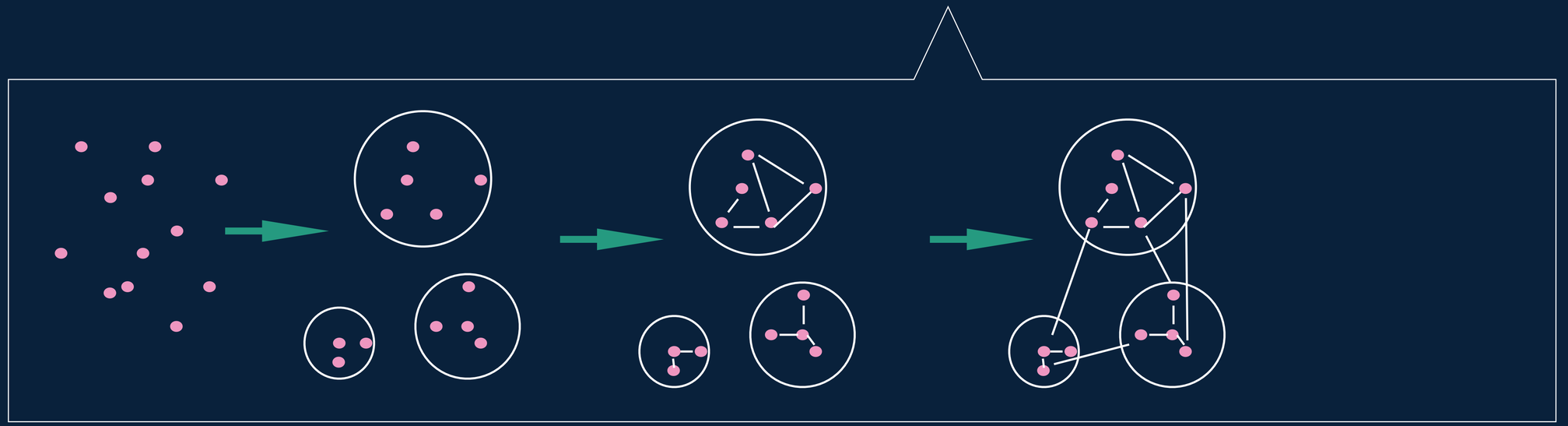
Measure



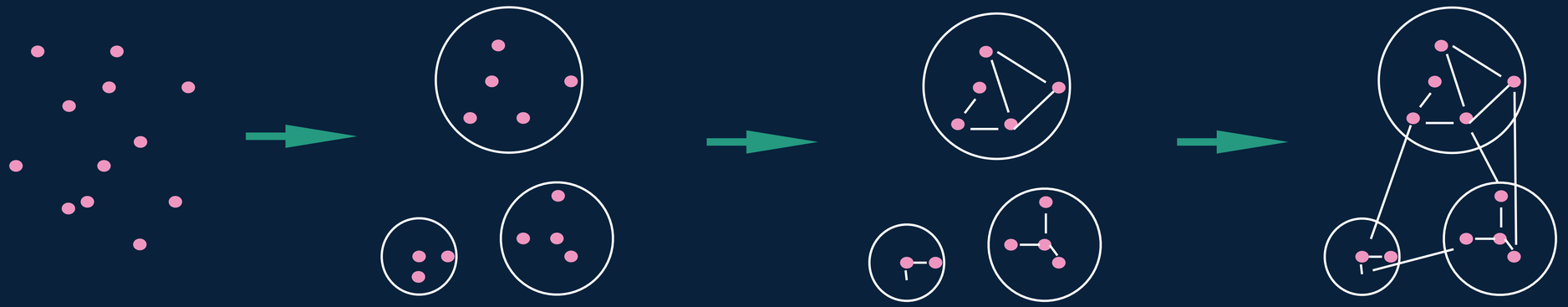
Darwini

Generated Graph





Darwini шаг за шагом



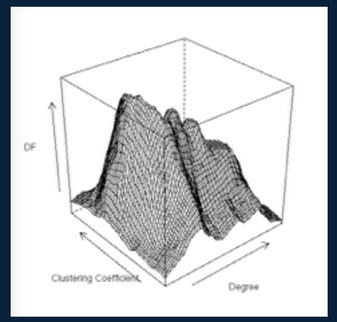
Создать вершины
Каждой вершине назначит
желаемое количество ребер
и локальный коэффициент
кластеризация

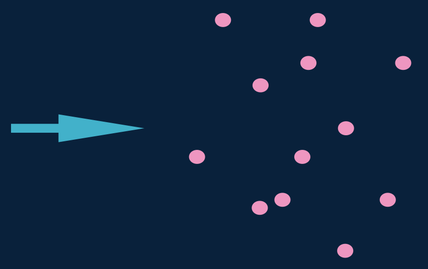
Объединить вершины, которые
должны учавствовать в одинаковом
количестве треугольников вместе

Добавить случайные ребра
внутри групп

Добавить случайные ребра между
группами

Darwini: создание вершин

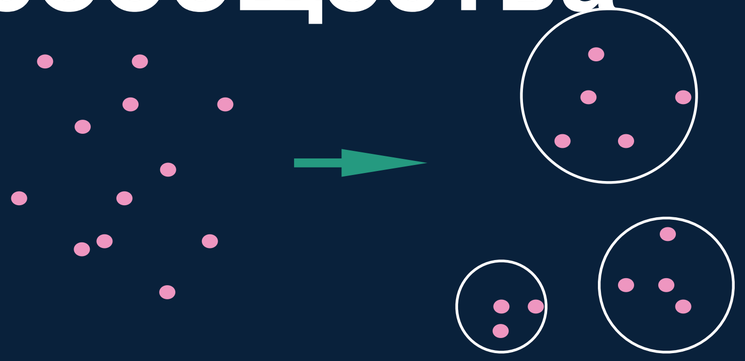




$$\forall c_i, d_i$$

Создать N вершин и “бросив монету” для каждой определить степень вершины и локальный коэффициент кластеризации

Darwini: объединение вершин в сообщества



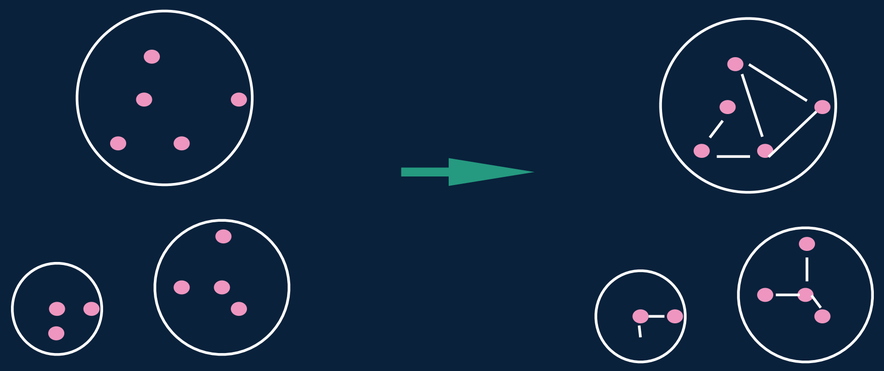
$$c_{e,i} = c_i d_i (d_i - 1)$$

Вершины, объединяемые в сообщество, будут учавствовать в одинаковом количестве треугольников

Ограничим размер каждого сообщества так, чтобы не превысить максимальное количество ребер

$$n \leq \min_{i \in B}(d_i) + 1 = n_{B,max}$$

Darwini: создание треугольников

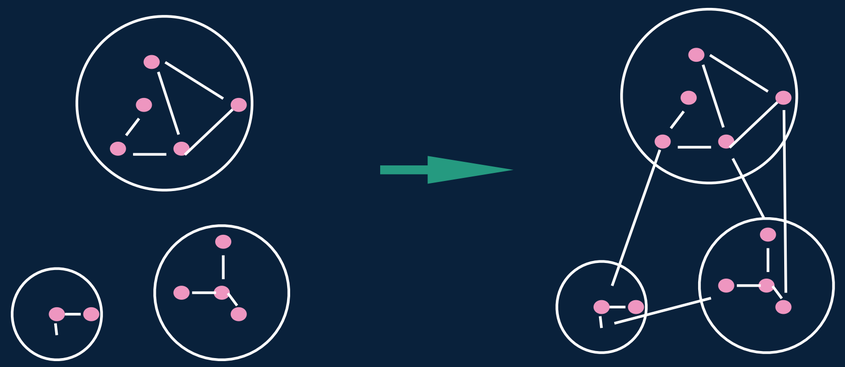


Ребра создаются случайно между каждой парой вершин, с вероятностью:

$$P_e = \sqrt[3]{\frac{c_i d_i (d_i - 1)}{(n-1)(n-2)}}$$

После этого шага, мы будем иметь достаточное количество треугольников, чтобы получить правильный коэффициент кластеризации

Darwini: создание ребер между сообществами



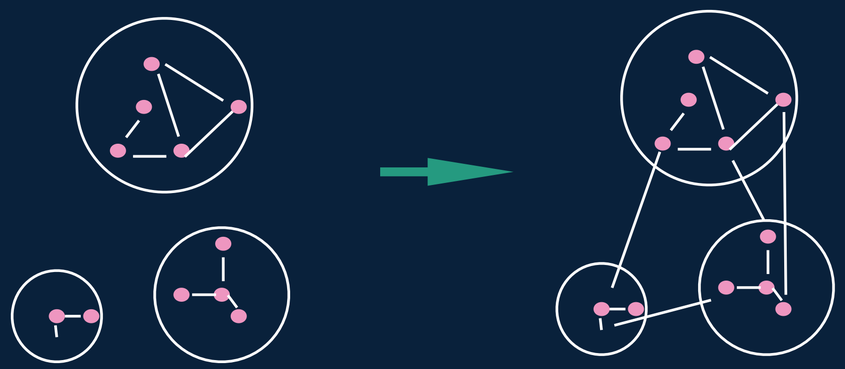
Для каждой вершины, выбираться другая случайная вершина. И если у другой вершины не хватает ребер, то между ними создается ребро

Проблема: тяжело найти партеров для вершин с большим количеством ребер

Добавление случайных ребер в Apache Giraph

1. Не вся информация доступна на каждом компьютере, получение информации может потребовать отправки сообщения
2. Выполнение должно происходить в параллель
3. Точное количество ребер не всегда необходимо
4. Чисто случайное добавление ребер искажает распределение степеней смежных вершин

Darwini: создание ребер для вершин с высокими степенями



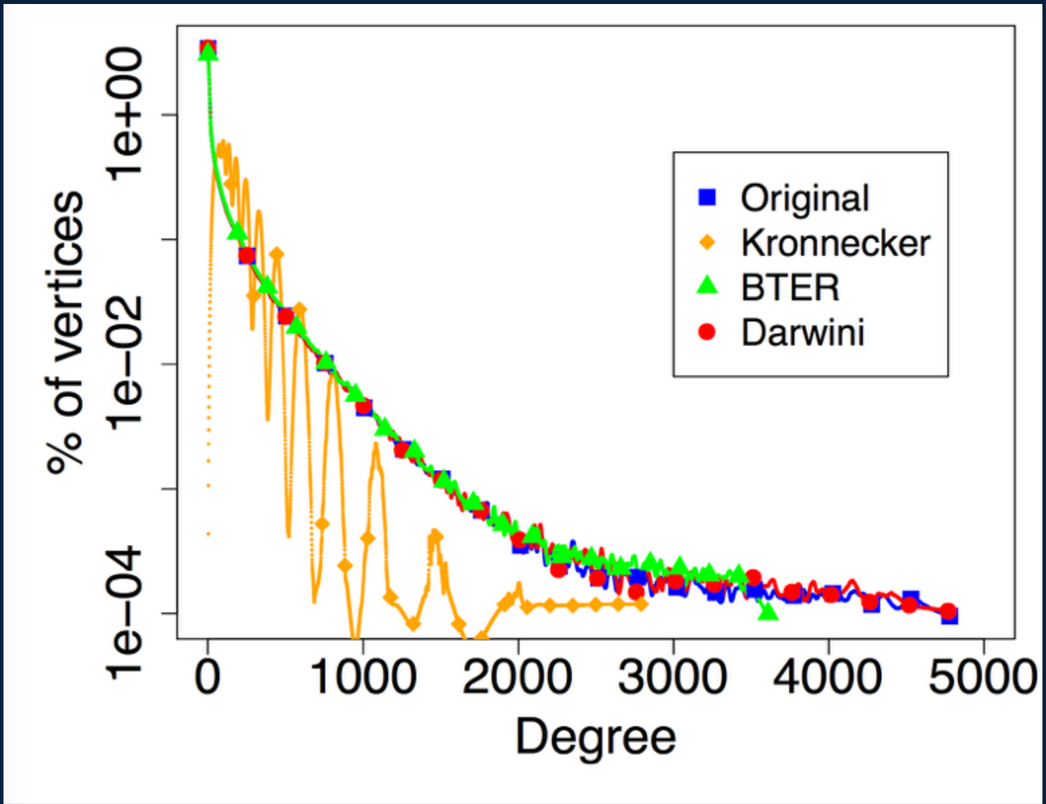
Рассматриваются только вершины у которых не достаточно ребер. Что гарантирует ограниченный размер группы

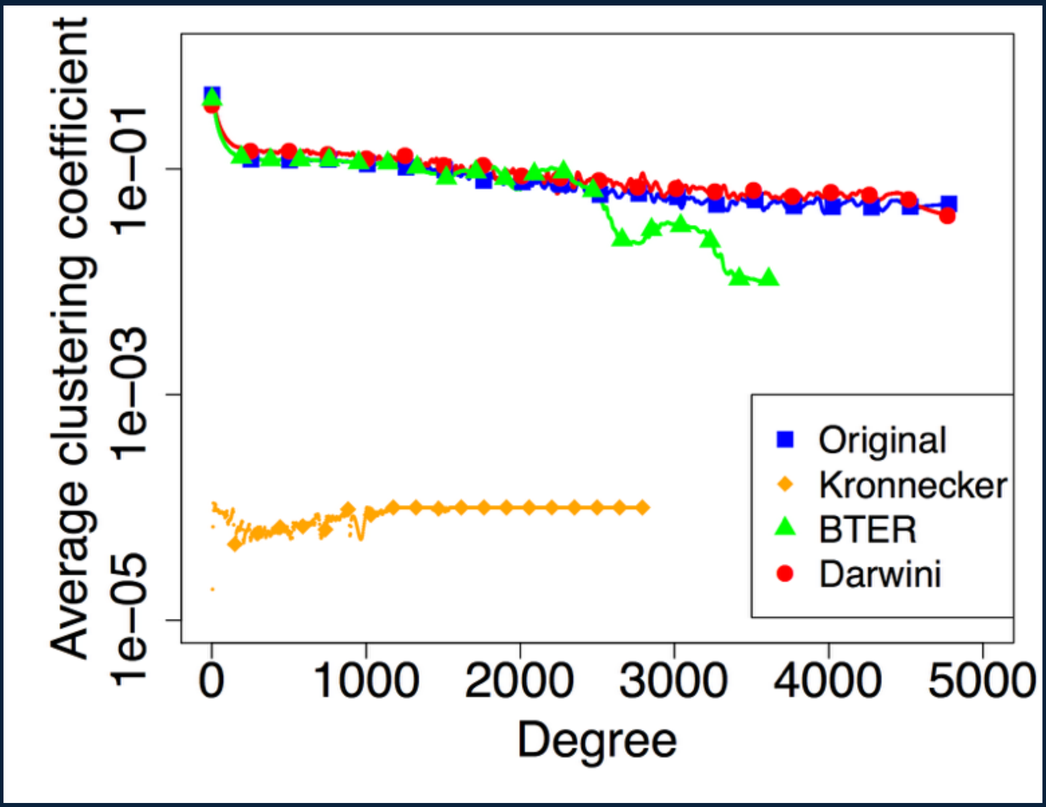
1. В каждой итерации вершины случайным образом объединяются во все более и более крупные группы
2. Для каждой пары вершин внутри группы ребро создается с вероятностью:

$$p = \frac{|d[i] - d[j]|}{d[i] + d[j]}$$

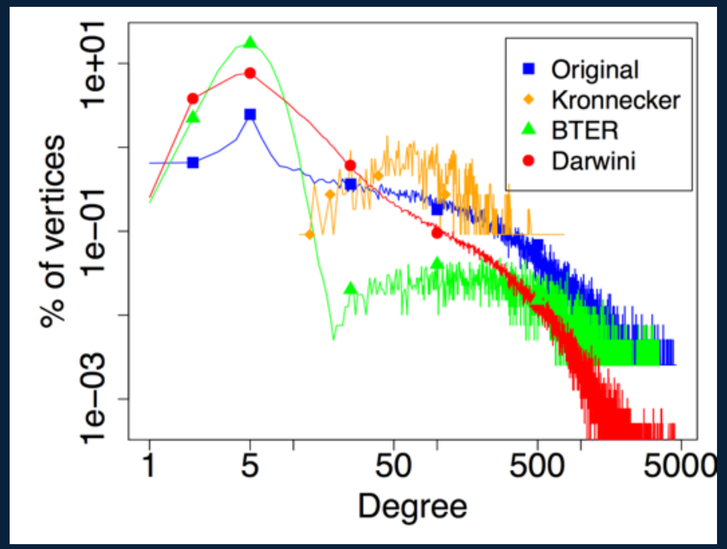
Результаты

Результаты: распределения степеней и коэффициента кластеризация

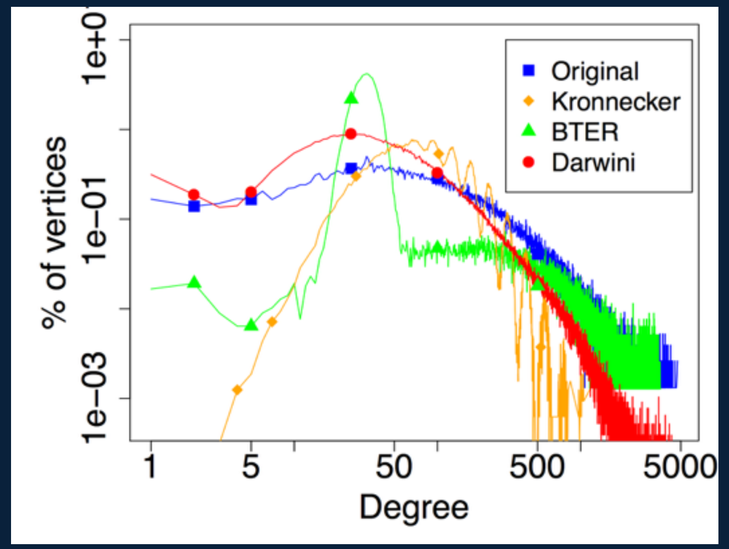




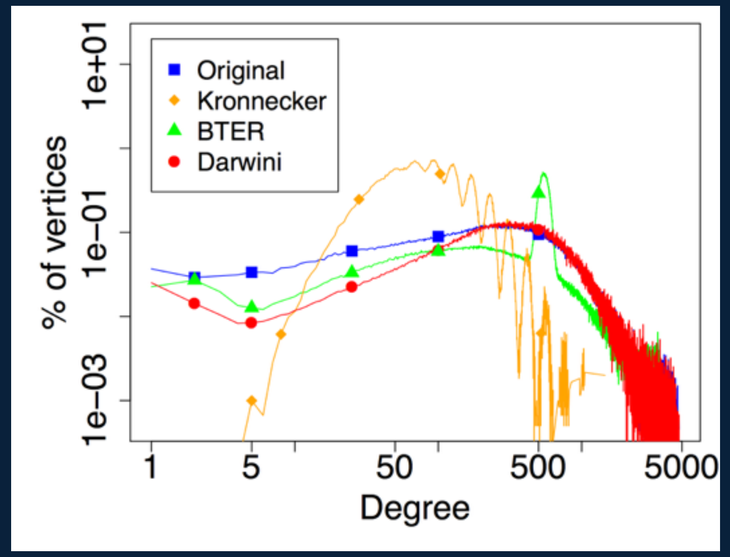
Результаты: распределение степеней смежных вершин



degree = 5

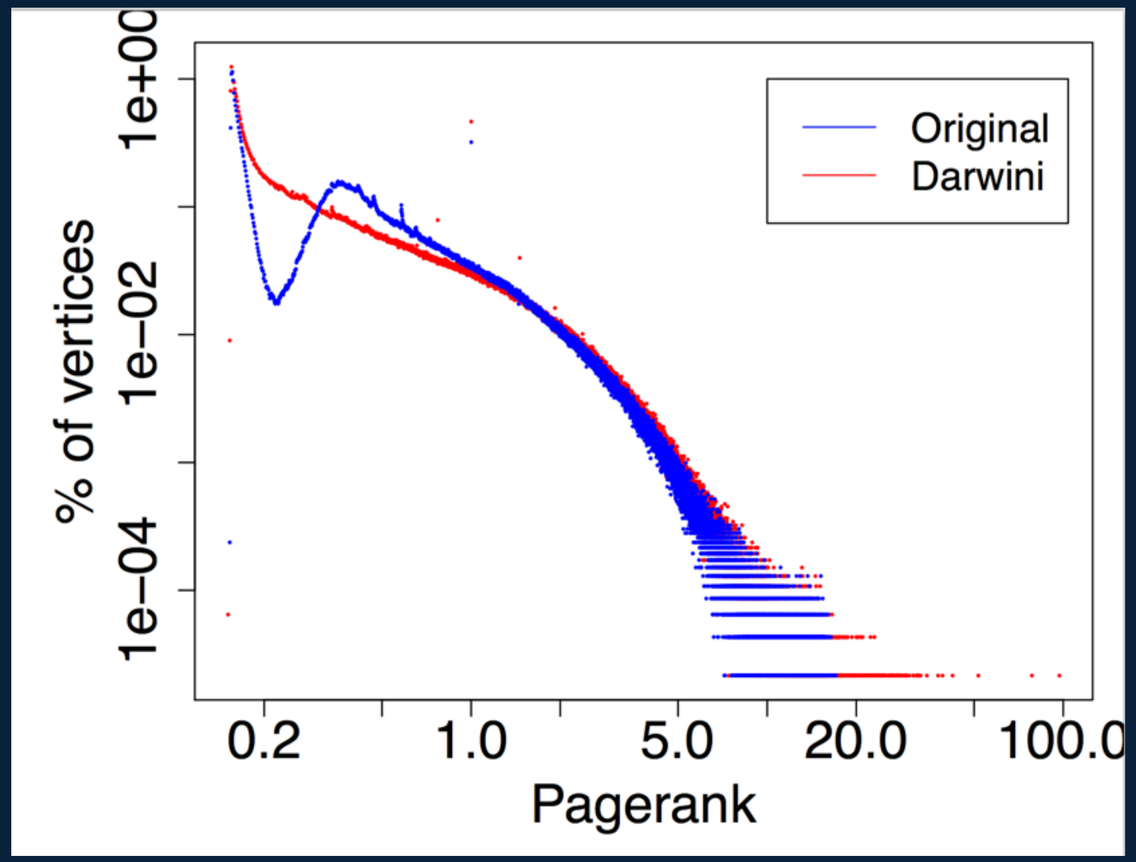


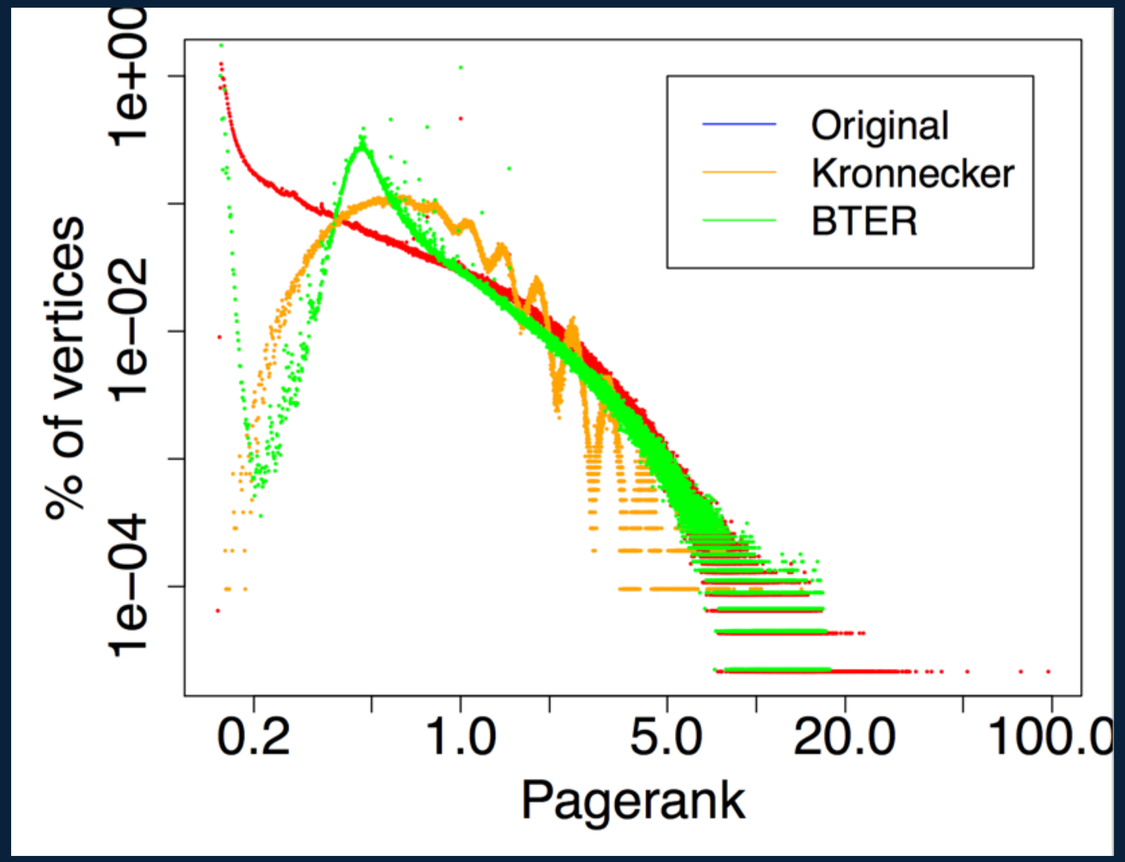
degree = 32



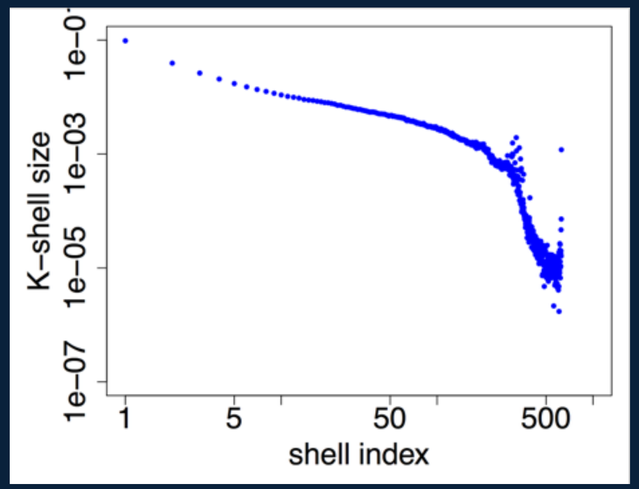
degree = 500

Результаты: page rank

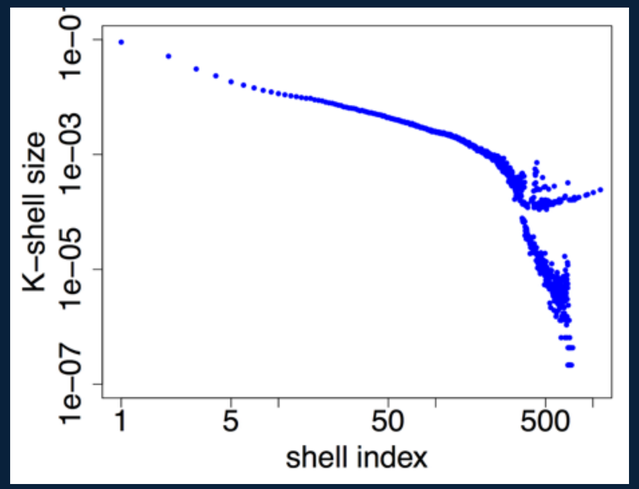




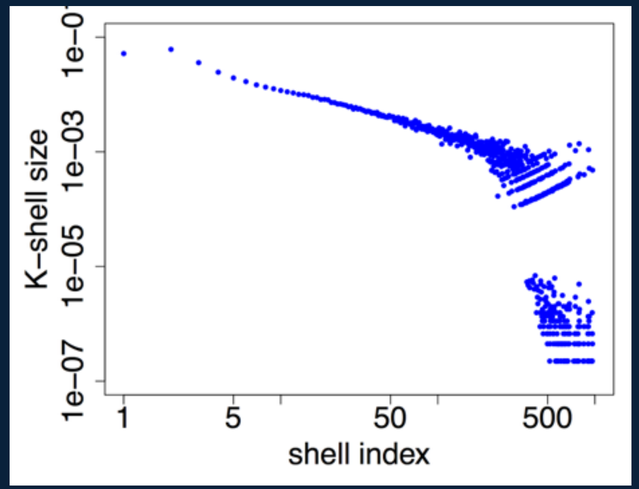
Результаты: K-Core декомпозиция



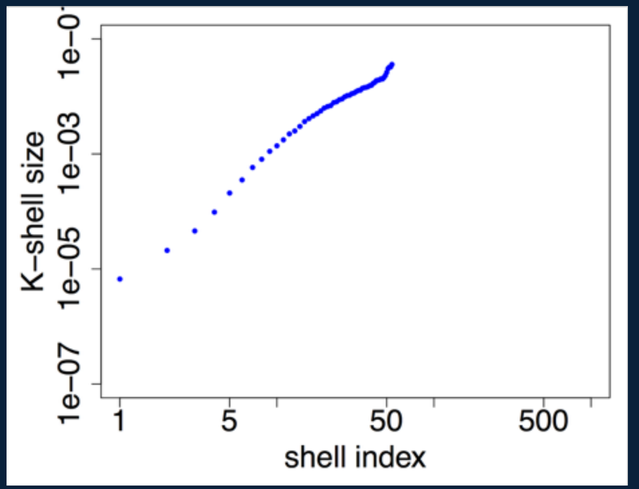
Original Graph



Darwini

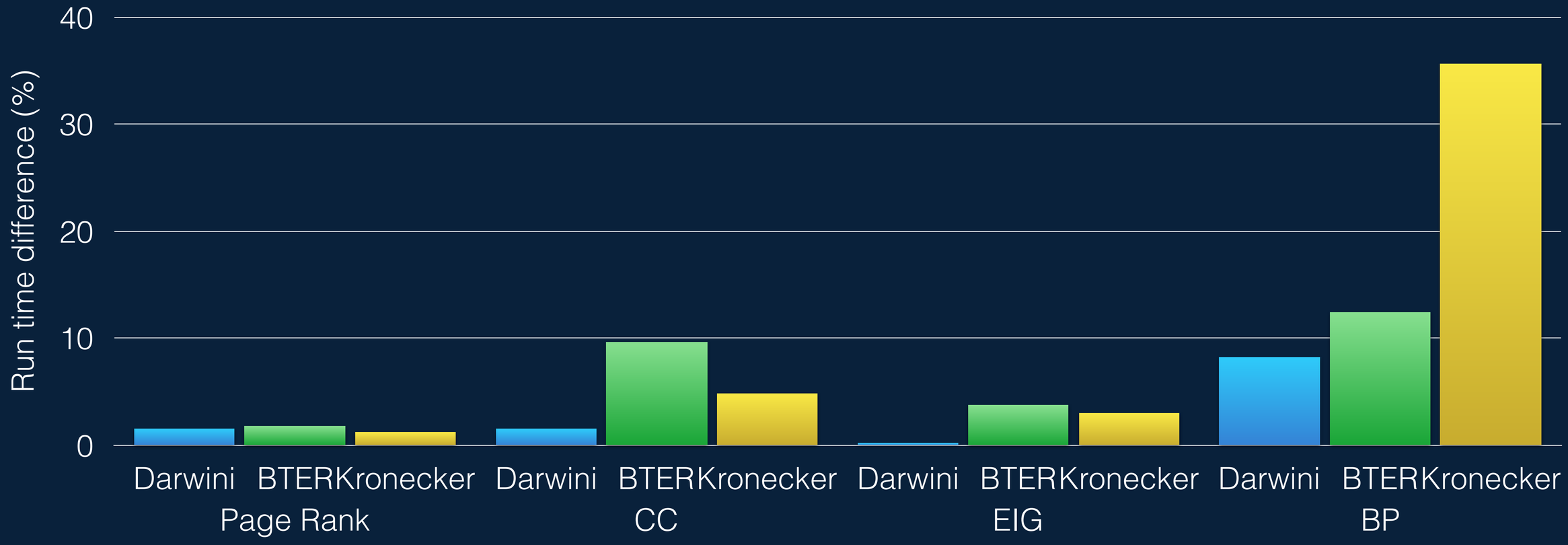


BTER

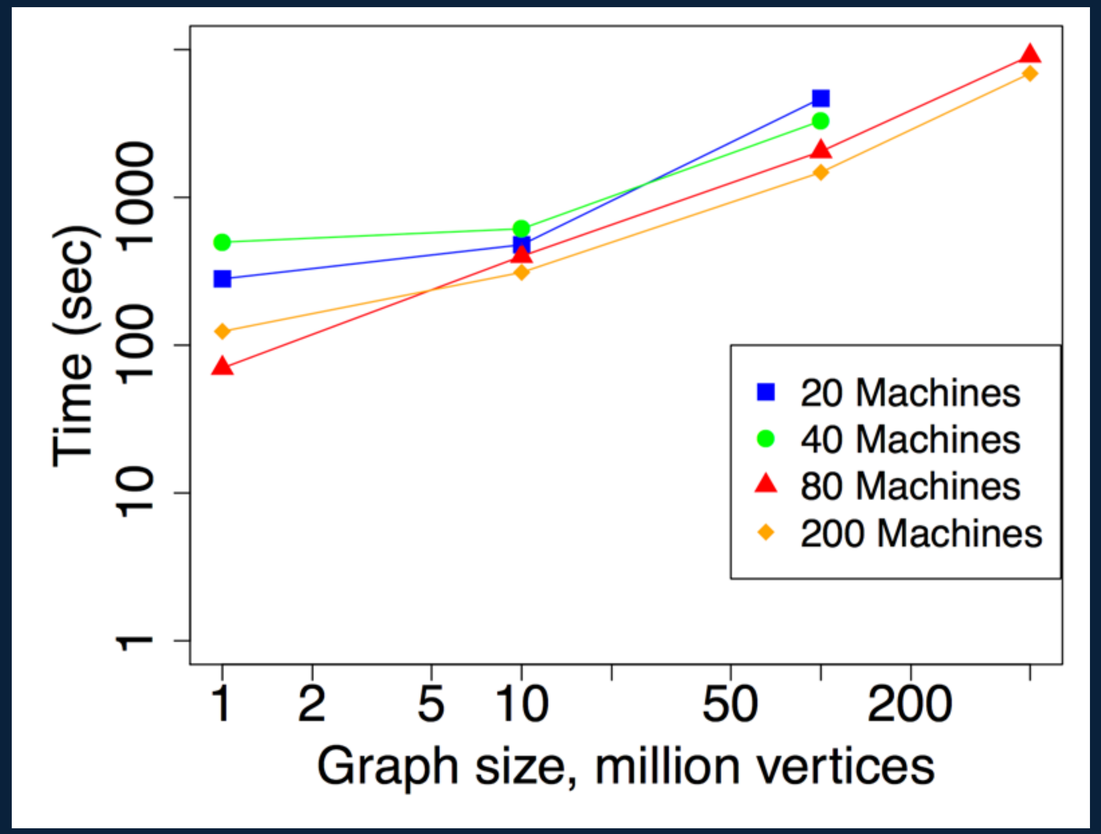


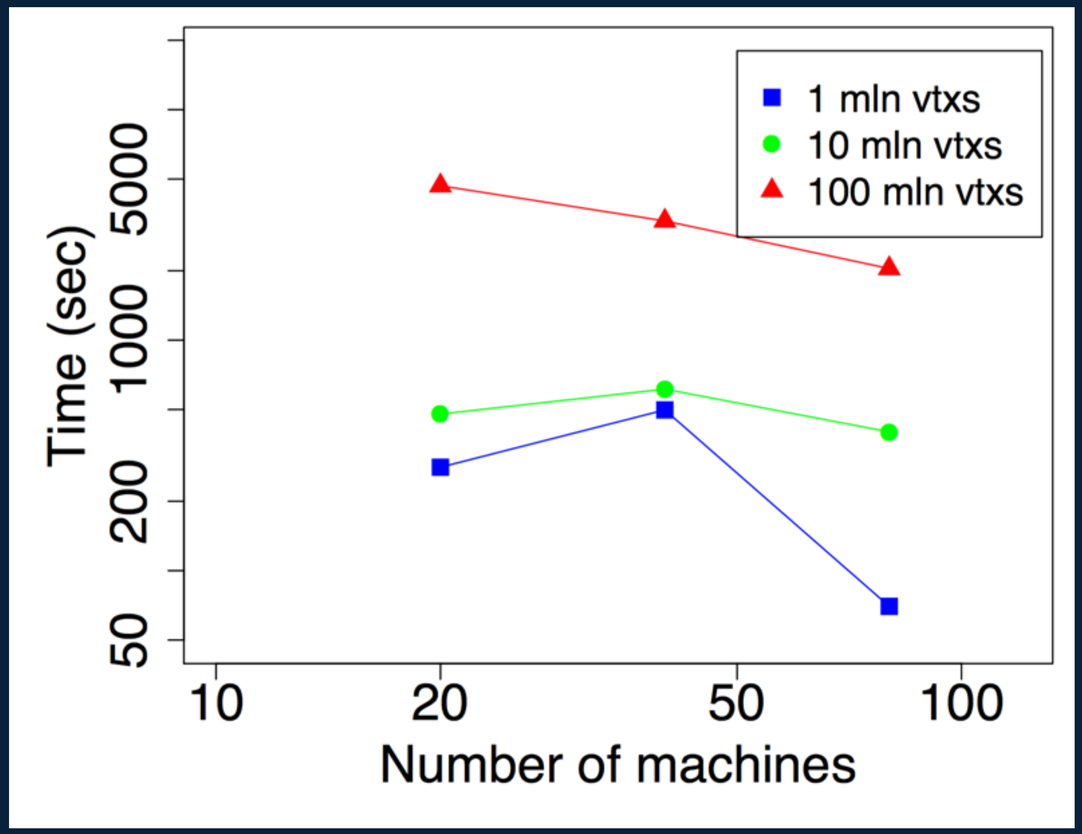
Kronecker

Результаты: производительность приложений



Darwini: скорость работы





Граф с триллионом ребер был создан за 7 часов

Спасибо за внимание!