Распределённая генерация случайных графов на основе моделей социальных сетей

Институт системного программирования РАН

Кирилл Чихрадзе et al.
chykhradze@ispras.ru

GraphHPC-2015

5 марта, 2015
Москва, Россия

План

- Немного о Spark
- Постановка задачи
- RGG: графы без сообществ
- CKB: графы с сообществами
- Тестирование
- Выводы

План

- Немного о Spark
- Постановка задачи
- RGG: графы без сообществ
- СКВ: графы с сообществами
- Тестирование
- Выводы

Что такое Spark?

- Независимая, быстрая платформа для распределённых вычислений, поддерживающая обработку данных по модели MapReduce, Pregel, GraphX
  - ✓ Хранение данных в оперативной памяти для быстрой обработки интерактивных запросов
  - ✓ Может быть в 100 раз быстрее Hadoop
- Совместим с системой хранения Hadoop (HDFS, Hbase, SequenceFiles и т. п.)

Модель программирования Spark

- Основная идея: сбое-устойчивые распределённые наборы данных (RDDs)
  - ✓ Распределённые коллекции объектов, которые могут быть кэшированы в памяти узлов кластера
  - ✓ Можно манипулировать с помощью различных параллельных операций (таких как map и reduce)
  - ✓ Автоматически перестраиваются при сбоях
- Интерфейс:
  - ✓ Элегантный интерфейс, интегрированный в язык Scala
  - ✓ Может использоваться интерактивно из консоли Scala

Пример: анализ логов

1. Загрузить сообщения с ошибками в память
2. Интерактивно выполнять запросы к ним

```
lines = spark.textFile("hdfs://...")
errors = lines.filter(_.startsWith("ERROR"))
messages = errors.map(_.split('\t')(2))
cachedMsgs = messages.cache()

cachedMsgs.filter(_.contains("foo")).count
cachedMsgs.filter(_.contains("bar")).count
. . .
```

Результат: запросы к 1 ТВ данных за 5-7с (против 170с для данных на диске)

План

- Немного о Spark
- Постановка задачи
- RGG: графы без сообществ
- СКВ: графы с сообществами
- Тестирование
- Выводы

Постановка задачи

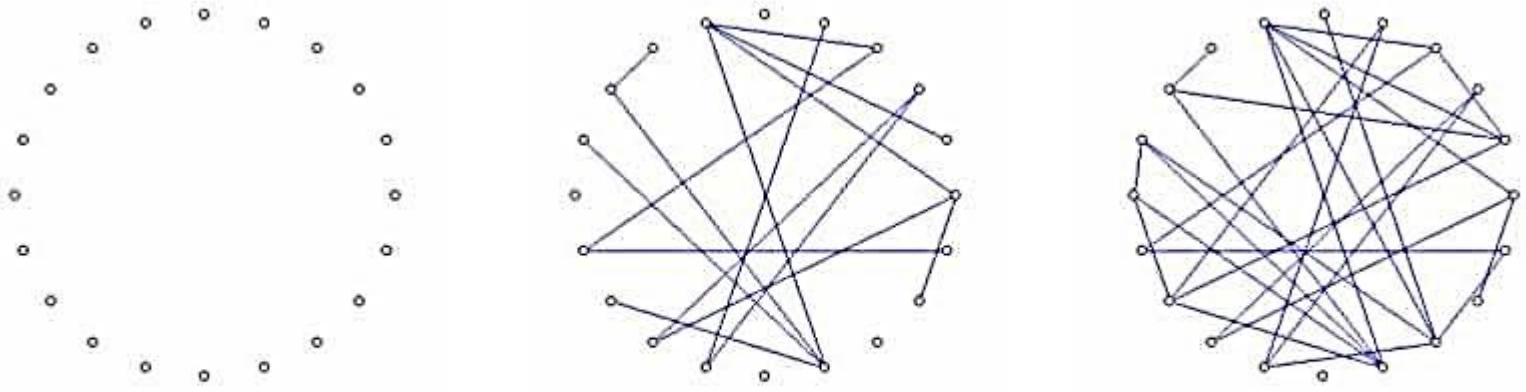
Сгенерировать случайный граф…

- …который будет удовлетворять основным свойствам социальных сетей;
- …за приемлемое время даже из миллиарда вершин;

Что такое случайный граф?

Граф Эрдёша-Реньи:

- N вершин
- Ребро появляется с вероятностью p

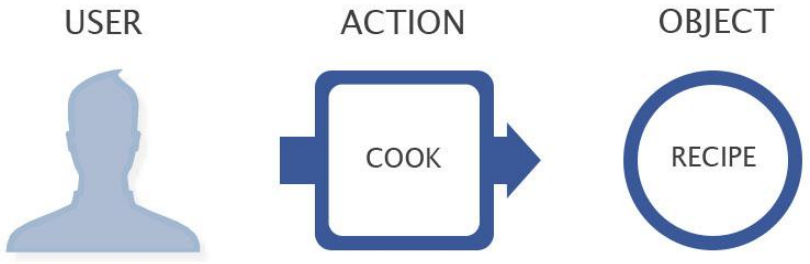


$p = 0$ (a) $p = 0.1$ (b) $p = 0.2$ (c)

Что такое социальный граф?

Типы узлов

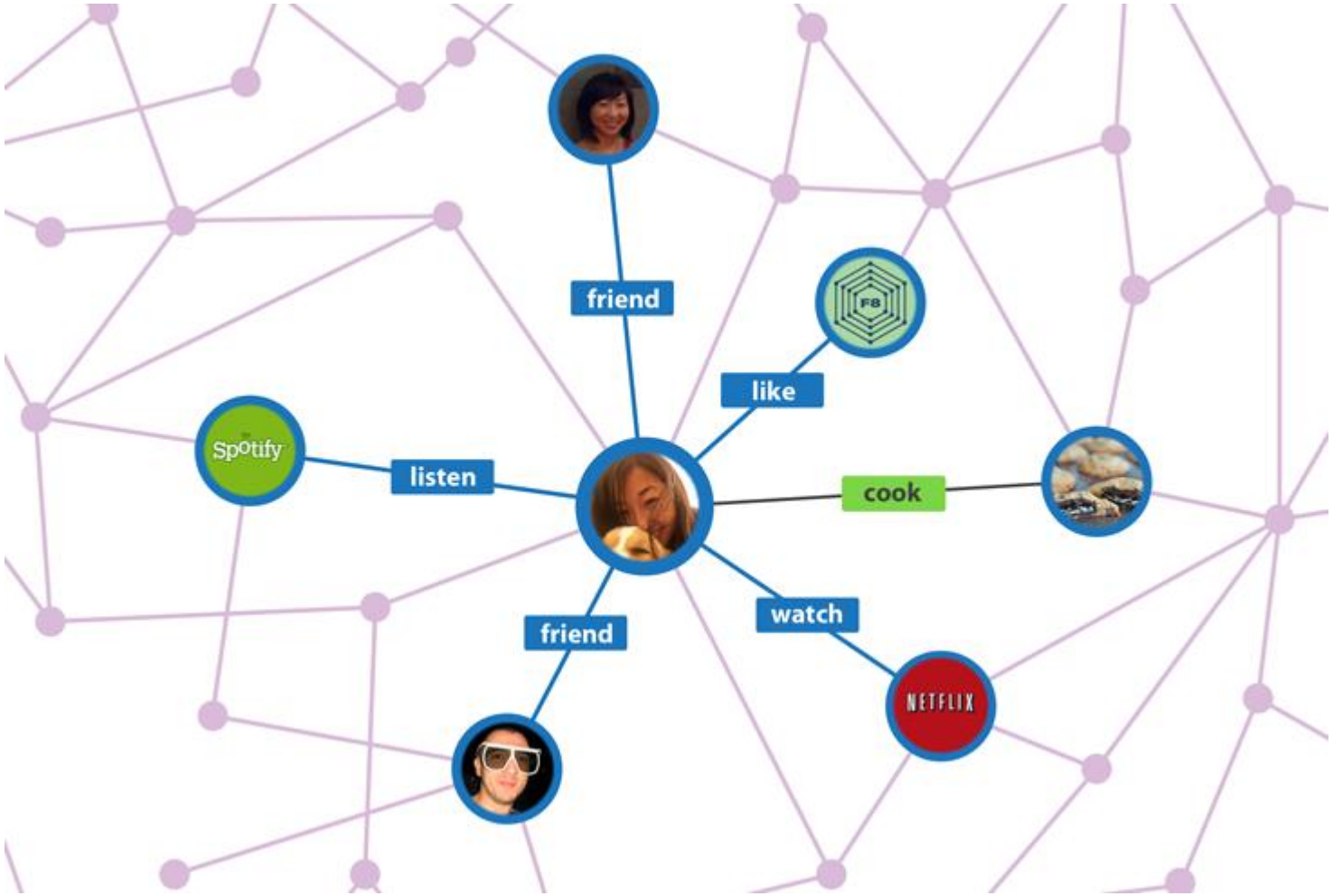
- *Пользователи*: поля профиля: атрибуты, интересы, контакты
- *Сообщества*: списки, группы
- *Контент*: сообщения, рисунки, видео



Типы рёбер

- *Социальные связи*: друзья, подписчики
- *Взаимодействие с контентом*: «лайки», репосты, комментарии

Что такое социальный граф?



Свойства социальных сетей

- Степени вершин распределены по степенному закону:

$$\mathbb{P}(\mathrm{x}) \propto x^{-\beta}$$

- Малый эффективный диаметр:

$$D \approx \frac{\ln(N)}{\ln(\ln(N))}$$

- Пользователи кластеризованы в пересекающиеся сообщества

Мотивация

- Размеры современных социальных сетей достигают сотен миллионов вершин.
- Необходимы алгоритмы анализа сетей, чья эффективность доказана на больших графах.
- Сбор реальных данных больших размеров затруднён вследствие временных и ресурсных затрат.

План

- Немного о Spark
- Постановка задачи
- RGG: графы без сообществ
- СКВ: графы с сообществами
- Тестирование
- Выводы

RGG: графы без сообществ

- Степени вершин распределены по степенному закону:

$$\mathbb{P}(\mathrm{x}) \sim x^{-\beta}$$

- Малый эффективный диаметр:

$$D \approx \frac{\ln(N)}{\ln(\ln(N))}$$

- Структура сообществ

RGG: входные параметры

- N_1 – количество вершин
- d_{mean} – средняя степень
- β – экспонента степенного распределения

RGG: основные шаги

1. Генерация натуральных чисел (степеней вершин) по степенному распределению

2. Подсчёт количества необходимых для генерации рёбер

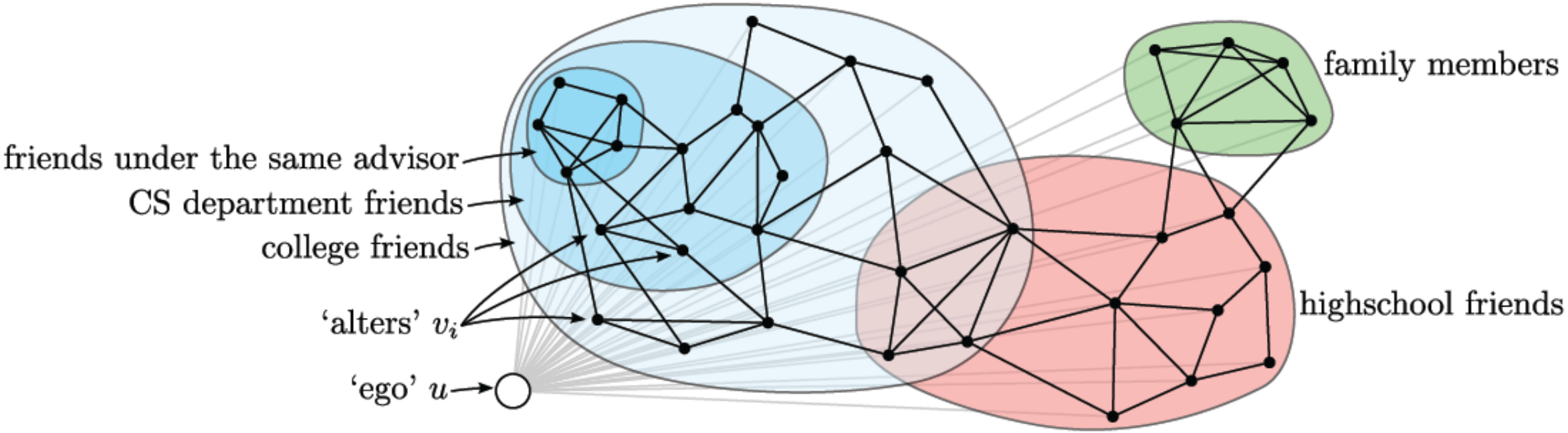
3. Выбор пар чисел — рёбер — (i, j) пропорционально их степеням

RGG: расширения

- Направленная реализация
- Генерация двудольного графа
- Атрибуты, тексты, «лайки»
- Сообщества

Что такое сообщество?

Что такое сообщество?



План

- Немного о Spark
- Постановка задачи
- RGG: графы без сообществ
- СКВ: графы с сообществами
- Тестирование
- Выводы

СКВ: графы с сообществами

- Степени вершин распределены по степенному закону:

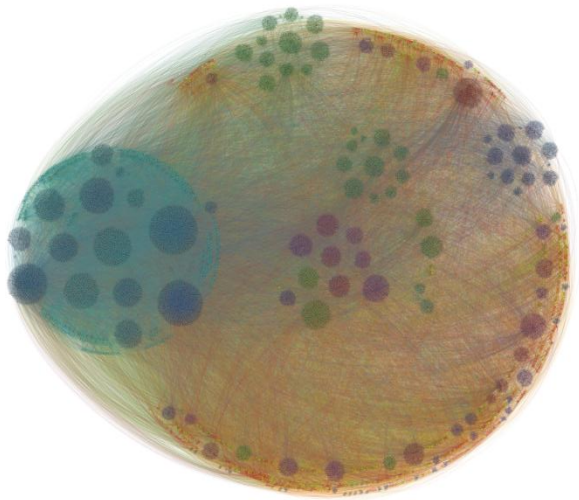
$$\mathbb{P}(\mathrm{x}) \sim x^{-\beta}$$

- Малый эффективный диаметр:

$$D \approx \frac{\ln(N)}{\ln(\ln(N))}$$

- Структура сообществ

СКВ: входные параметры

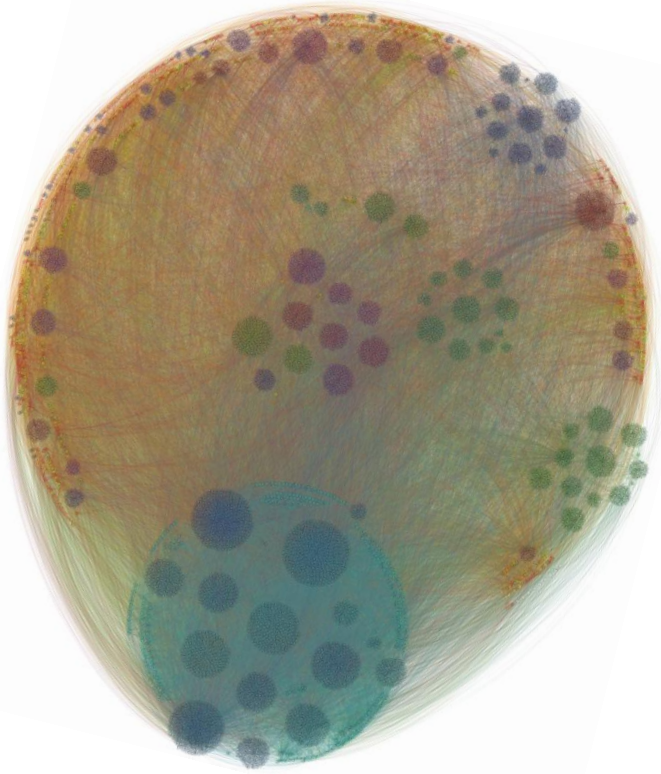


- N_1 – количество вершин
- d_{mean} – средняя степень
- Параметры степенного распределения: минимальное и максимальное значения, показатель степени
- α, γ – две константы, определяющие вероятность ребра
- ε – вероятность ребра между пользователями безотносительно сообществ

СКВ

Основные шаги:

1. Генерация двудольного графа вершина-сообщество
2. Генерируются рёбра внутри сообществ
3. Генерируются рёбра между пользователями безотносительно сообществ

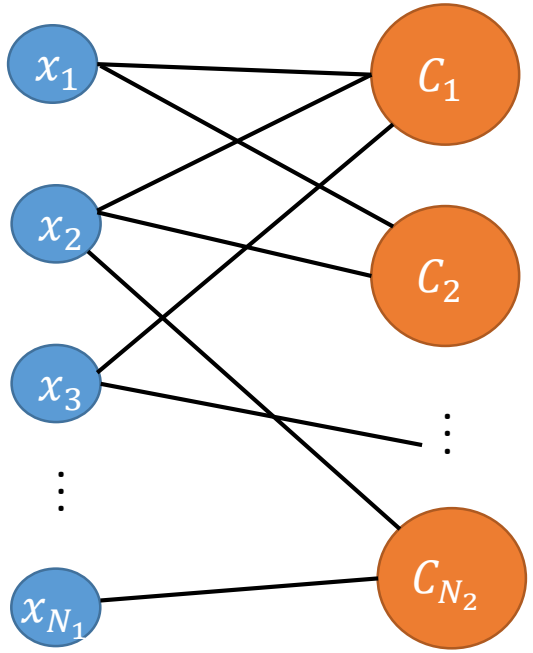


СКВ: вершина-сообщество

- Количество сообществ N_2 вычисляется из*

$$N_1 \cdot E[X_1] = N_2 \cdot E[X_2]$$

- Распределения количества вхождений вершин в сообщества (m_i) и размера сообществ (x_c) генерируется по степенному закону с экспонентами β_1 и β_2
- Создаётся графовая реализация 2 последовательностей степеней путём случайной попарной комбинации вершин из разных долей



*$E[X_1]$ и $E[X_2]$ – средние значения числа вхождений пользователя в сообщество и размера сообществ соответственно

СКВ: генерация рёбер

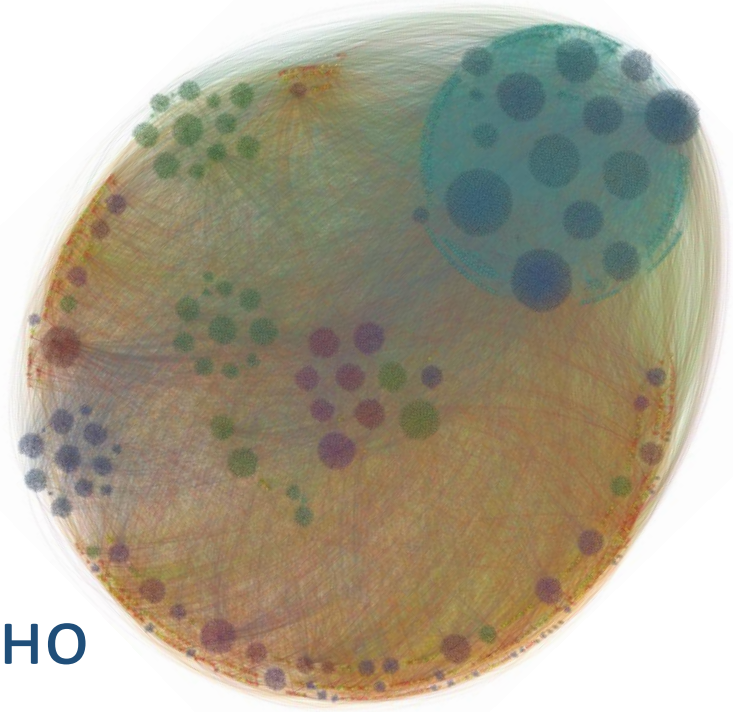
- Рёбра в сообществах генерируются с вероятностью*:

$$p_c = \frac{\alpha}{x_c^{\gamma}}$$

где x_c – размер сообщества

- Рёбра между пользователями безотносительно сообществ генерируются с вероятностью**:

$$p_{out} = \varepsilon$$



* Yang, J., and Leskovec, J. Structure and overlaps of communities in networks.
**Yang, J., and Leskovec, J. Community-affiliation graph model for overlapping network community detection.

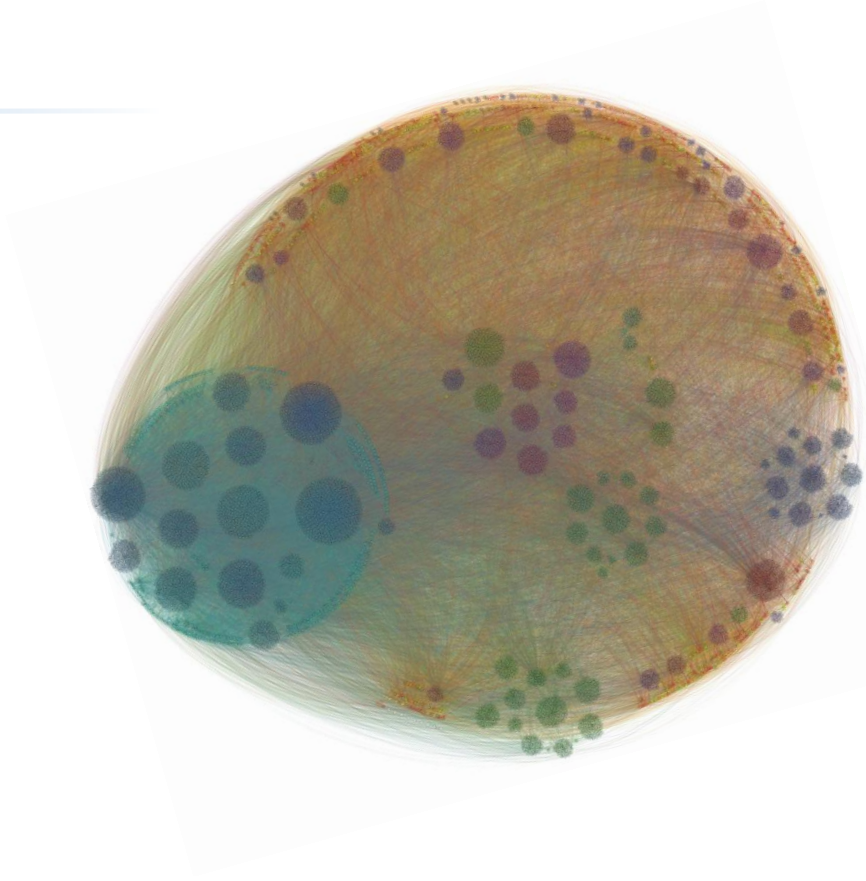
СКВ: генерация рёбер

- Количество рёбер в сообществе:

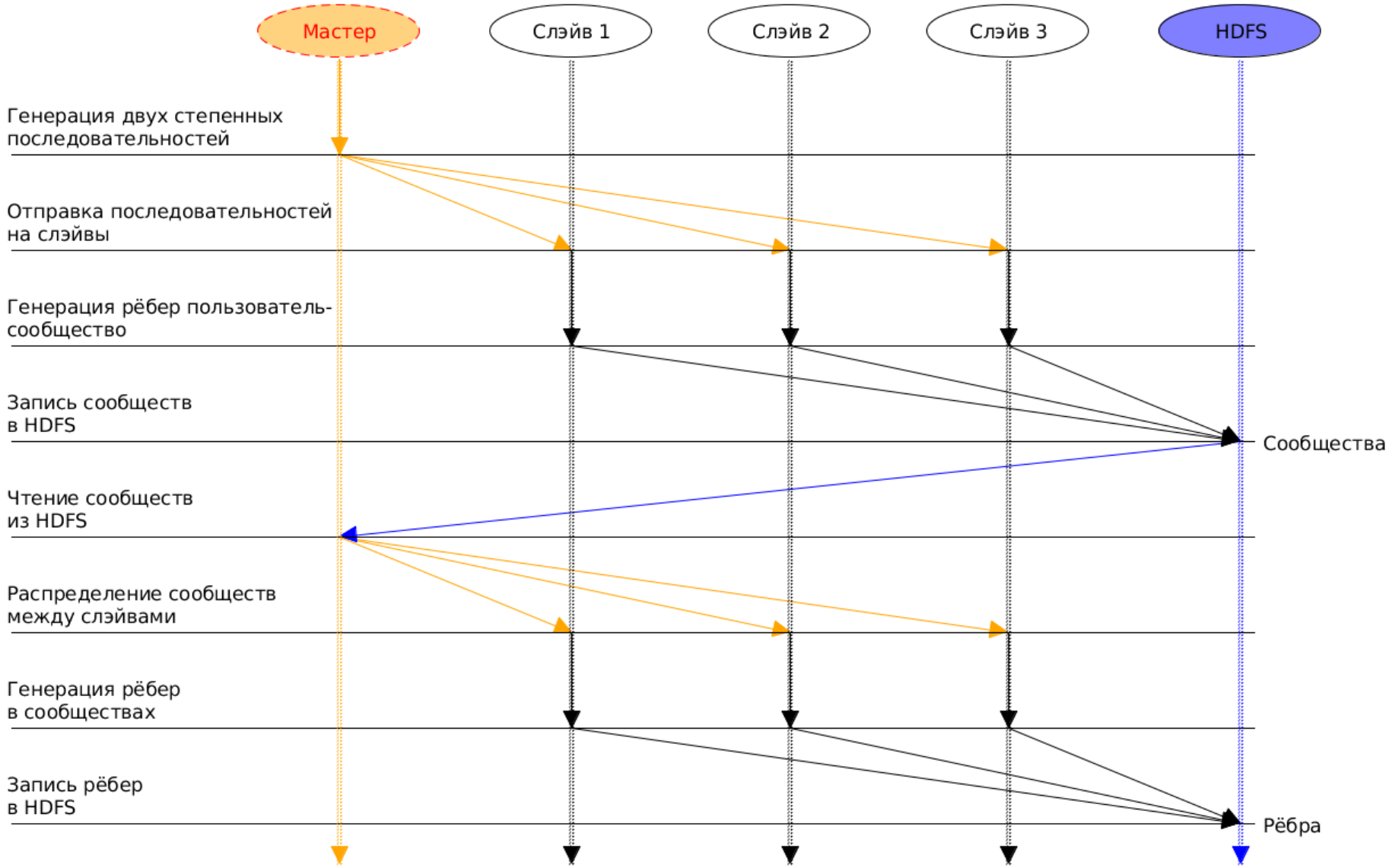
$$M_c \sim Bin\left(\frac{x_c(x_c-1)}{2}, \frac{\alpha}{x_c^{\gamma}}\right),$$

- Количество рёбер между пользователями безотносительно сообществ:

$$M_o \sim Bin\left(\frac{N_1(N_1-1)}{2}, \varepsilon\right)$$



СКВ: реализация Apache Spark



План

- Немного о Spark
- Постановка задачи
- RGG: графы без сообществ
- СКВ: графы с сообществами
- Тестирование
- Выводы

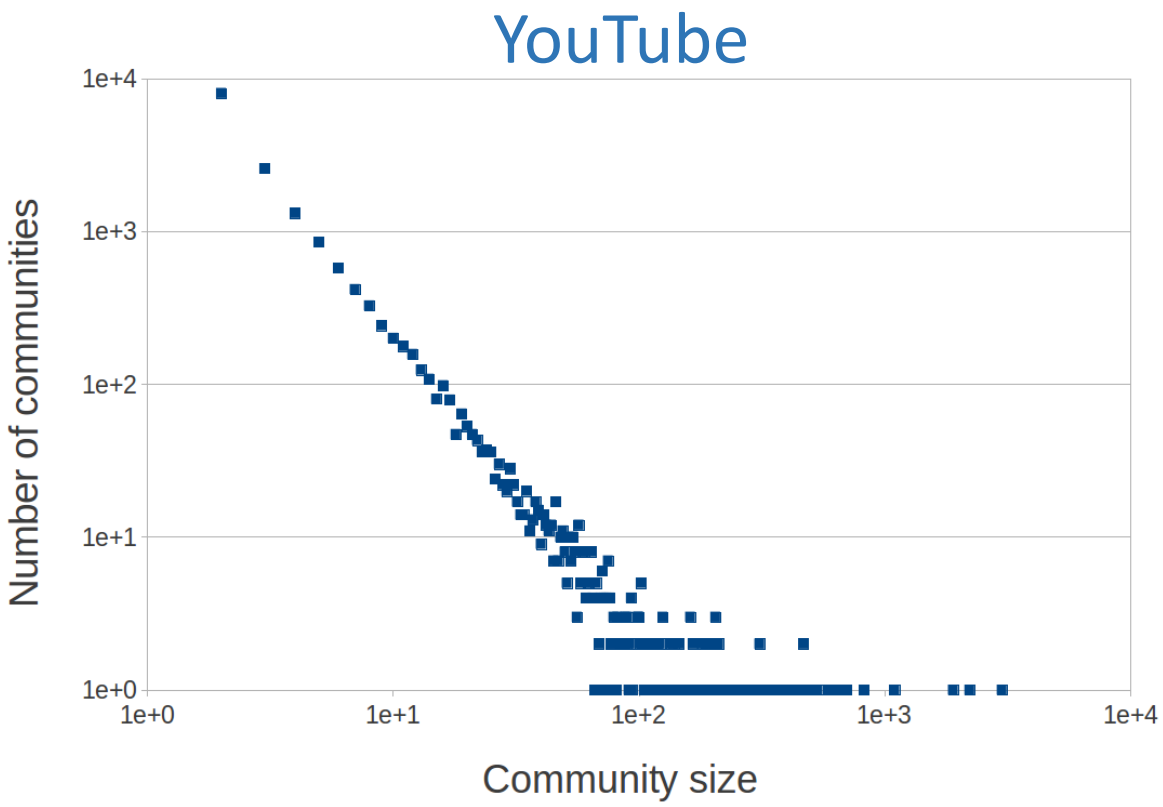
Сравнение с реальными данными

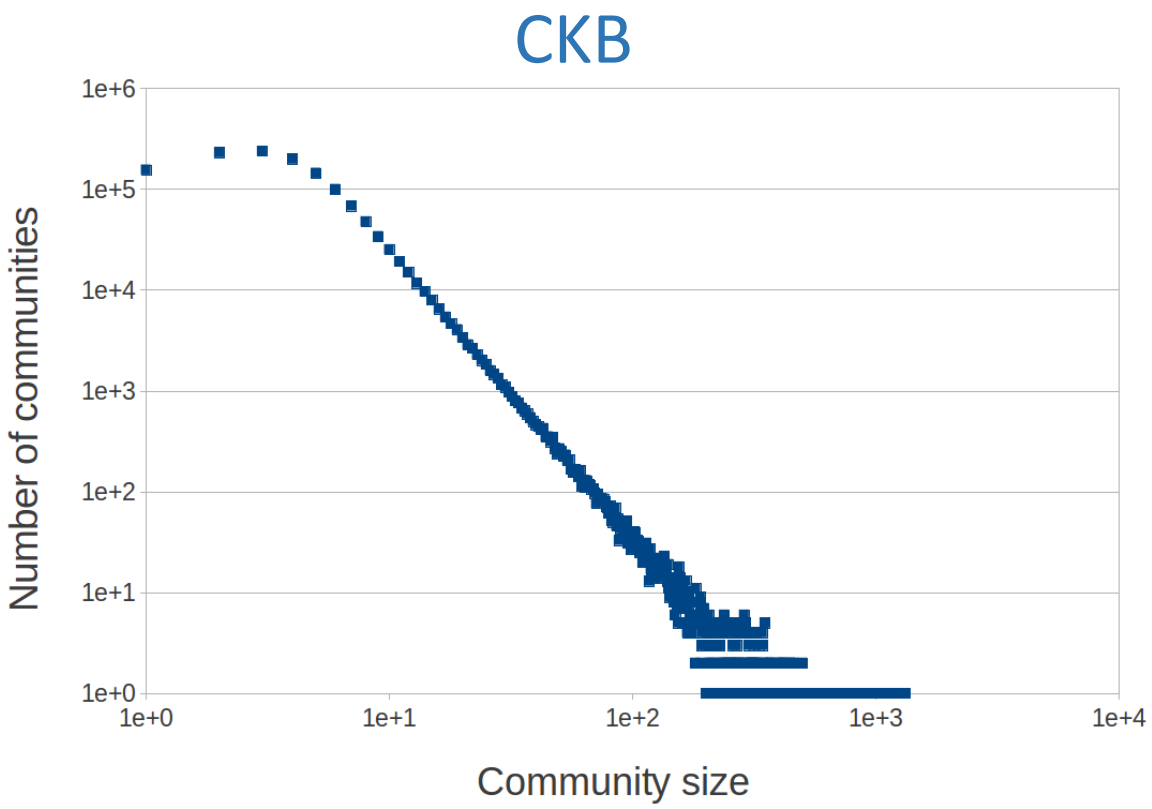
	LiveJournal	СКВ
Количество вершин	$\sim 4 \cdot 10^6$	$\sim 4.2 \cdot 10^6$
Количество рёбер	$\sim 34.6 \cdot 10^6$	$\sim 38.2 \cdot 10^6$
$\beta_{\text{вершины}}$	2.14	2.15
$\beta_{\text{сообщества}}$	2.22	2.26
$\beta_{\text{степени}}$	2.15	2.15
Медиана распределения размеров сообществ (x_c)	10	8
Медиана распределения количества вхождения вершин в сообщества (m_i)	2	2
Доля вершин с $m_i > 1$	63%	66%
Средний коэффициент кластеризации	0.3538	0.1034
Эффективный диаметр	6.4	5.16

Сравнение с реальными данными

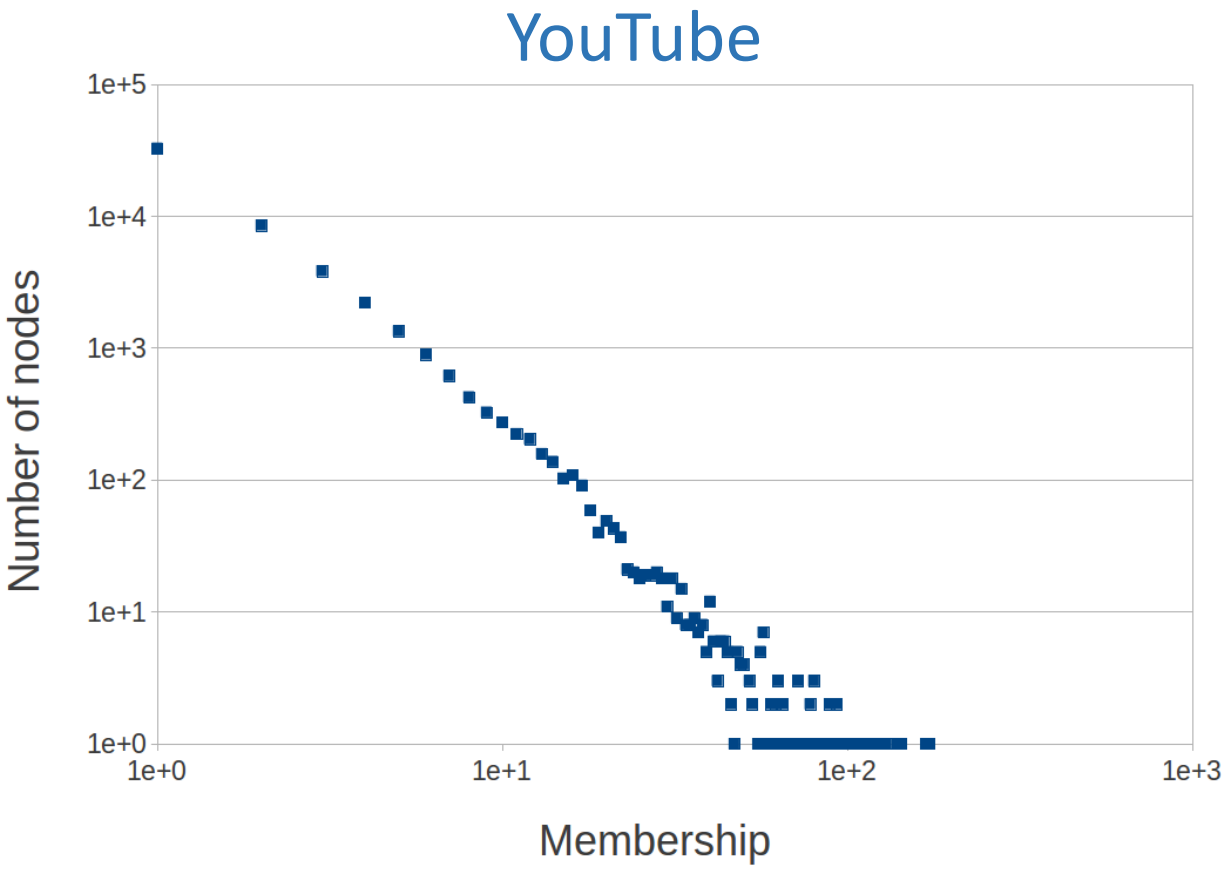
	YouTube	СКВ
Количество вершин	$\sim 1.1 \cdot 10^6$	$\sim 1.1 \cdot 10^6$
Количество рёбер	$\sim 3 \cdot 10^6$	$\sim 3 \cdot 10^6$
$\beta_{\text{вершины}}$	2.36	2.41
$\beta_{\text{сообщества}}$	2.83	2.95
$\beta_{\text{степени}}$	2.53	2.45
Медиана распределения размеров сообществ (x_c)	3	4
Медиана распределения количества вхождения вершин в сообщества (m_i)	2	2
Доля вершин с $m_i > 1$	38%	68%
Средний коэффициент кластеризации	0.1723	0.1066
Эффективный диаметр	6.5	6.2

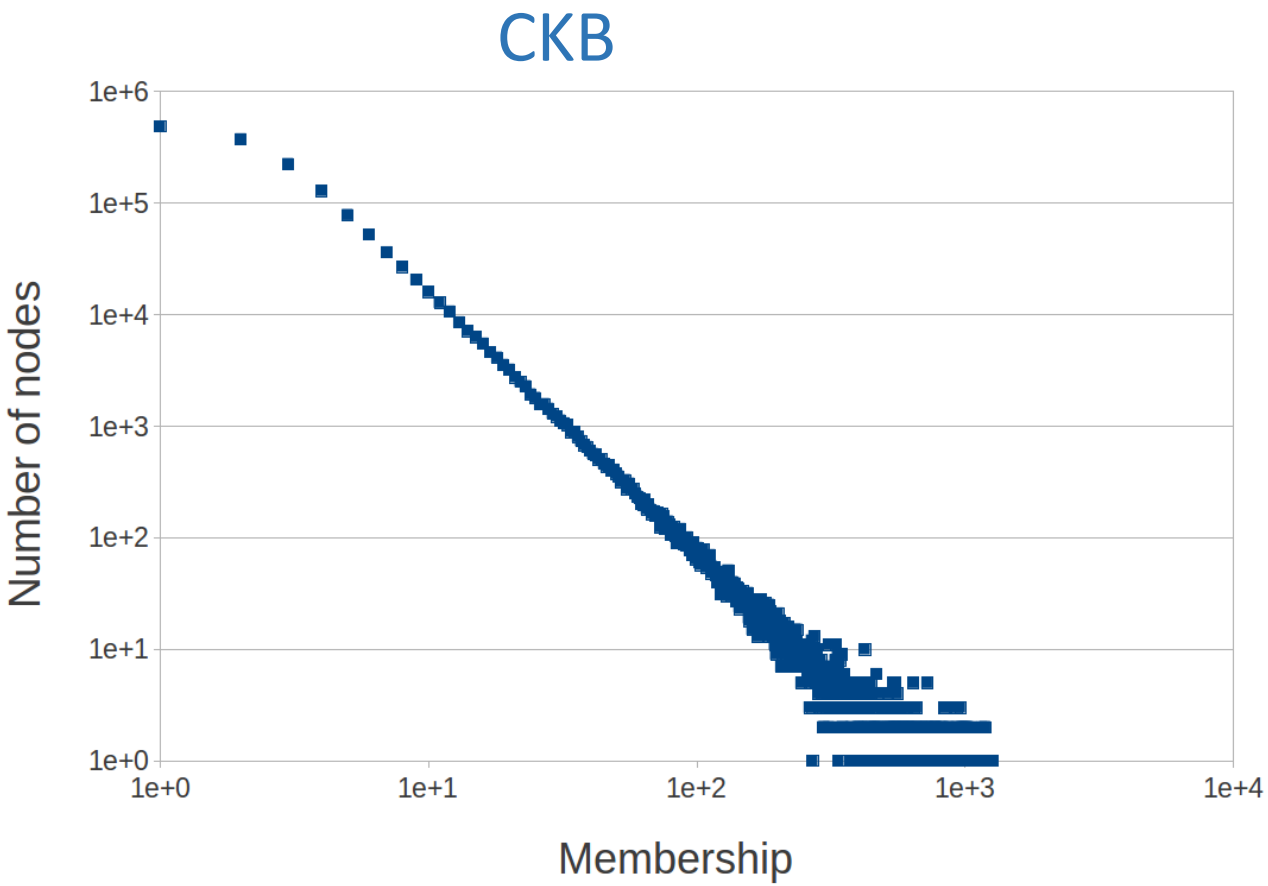
Сравнение YouTube и СКВ





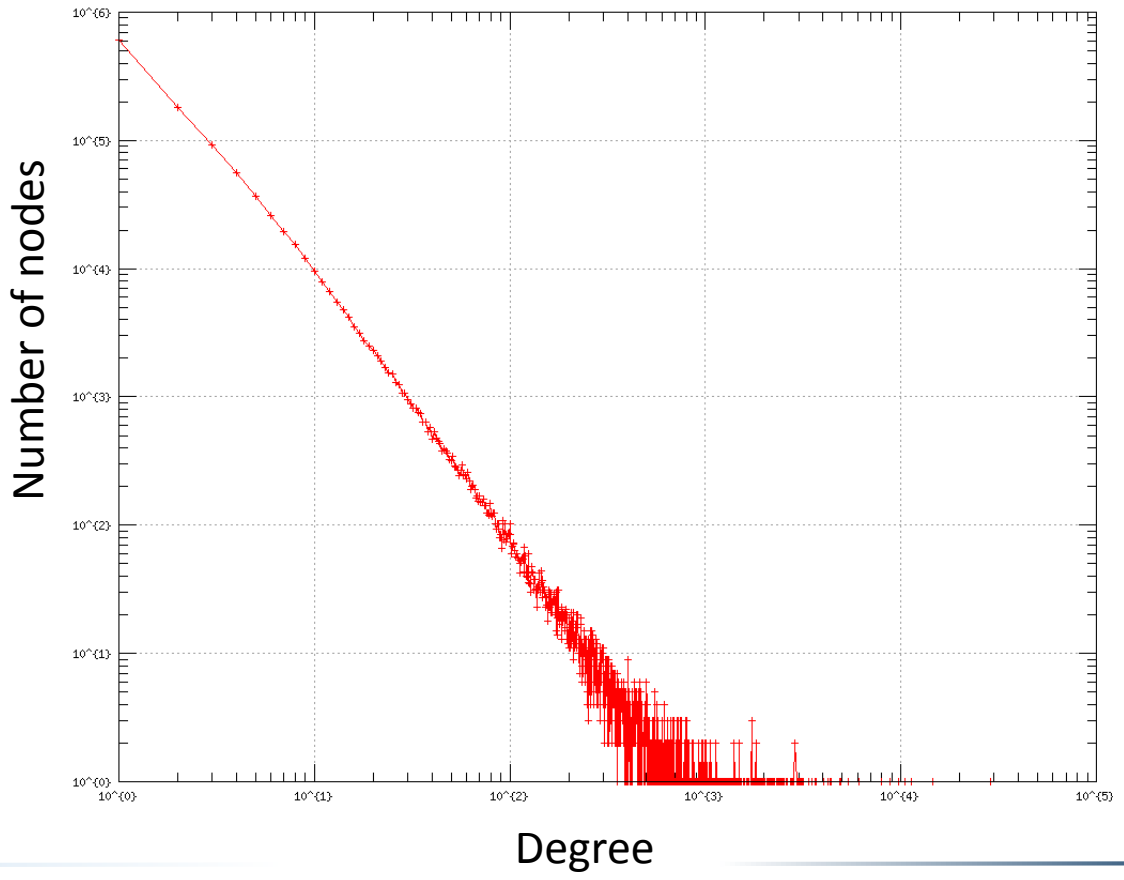
Сравнение YouTube и СКВ



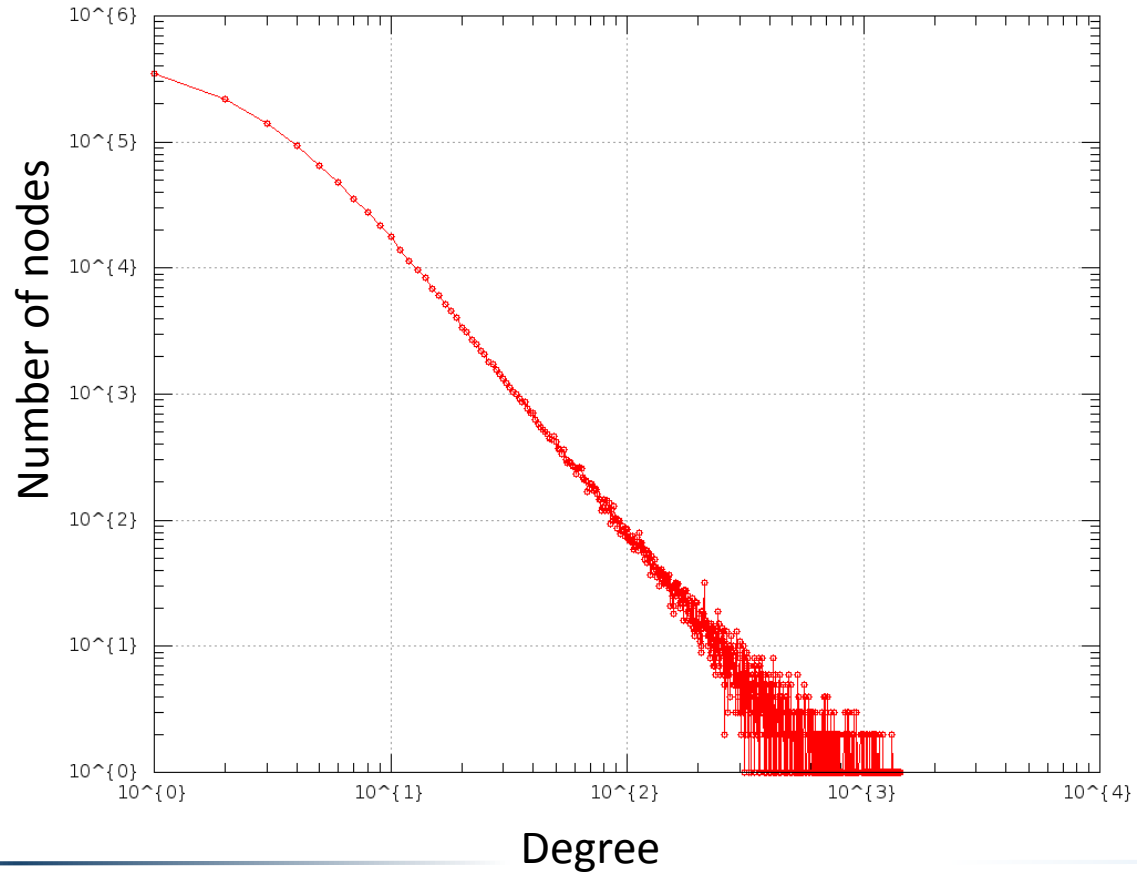


Сравнение YouTube и СКВ

YouTube

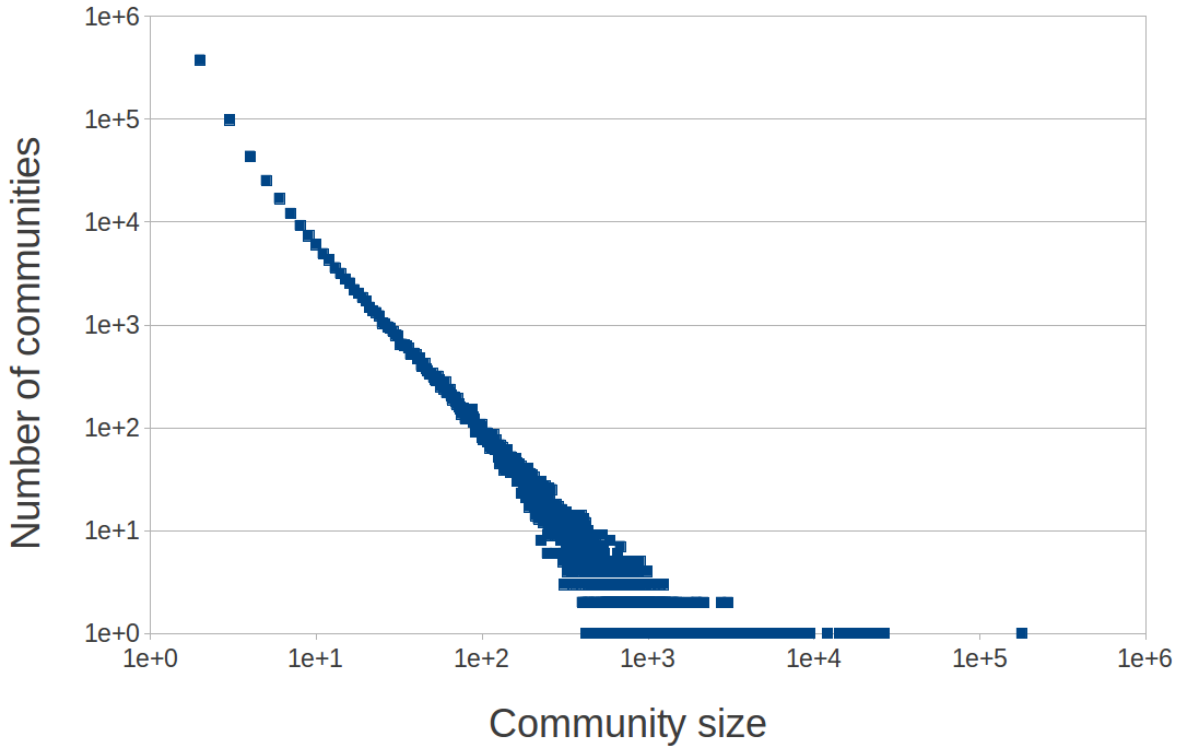


СКВ

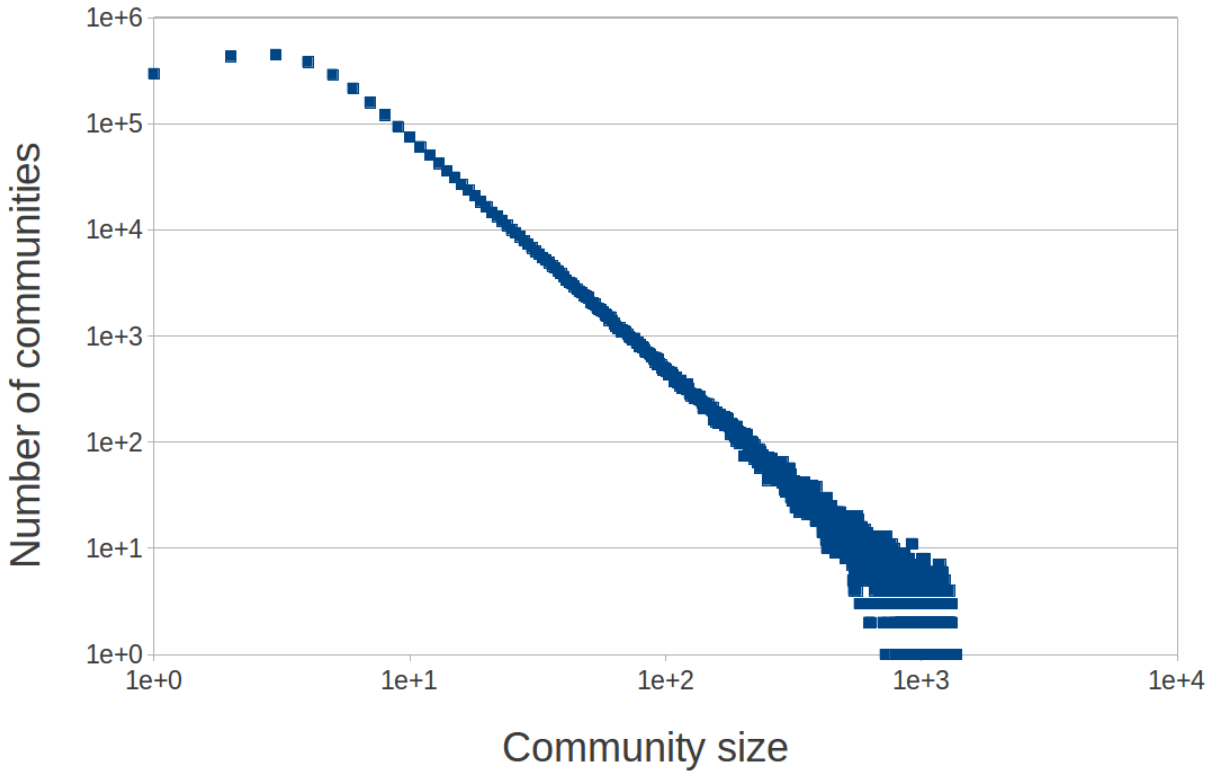


Сравнение LiveJournal vs. СКВ

LiveJournal

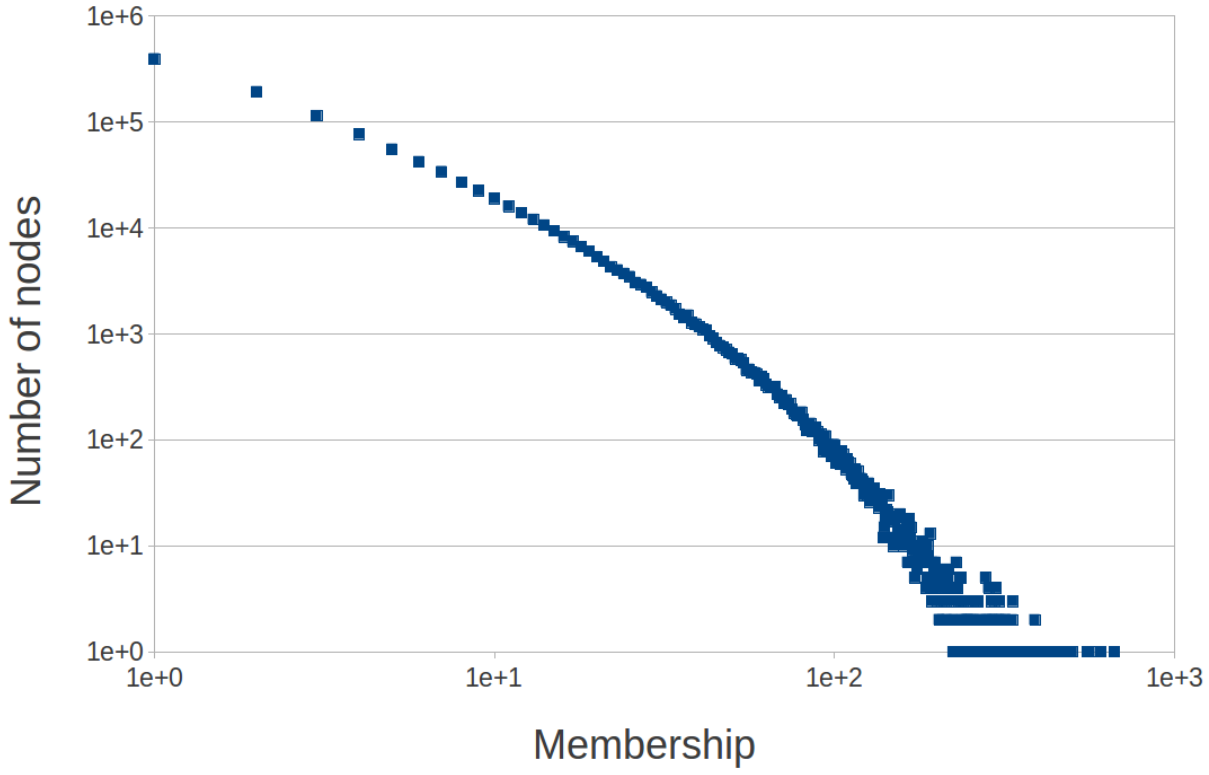


СКВ

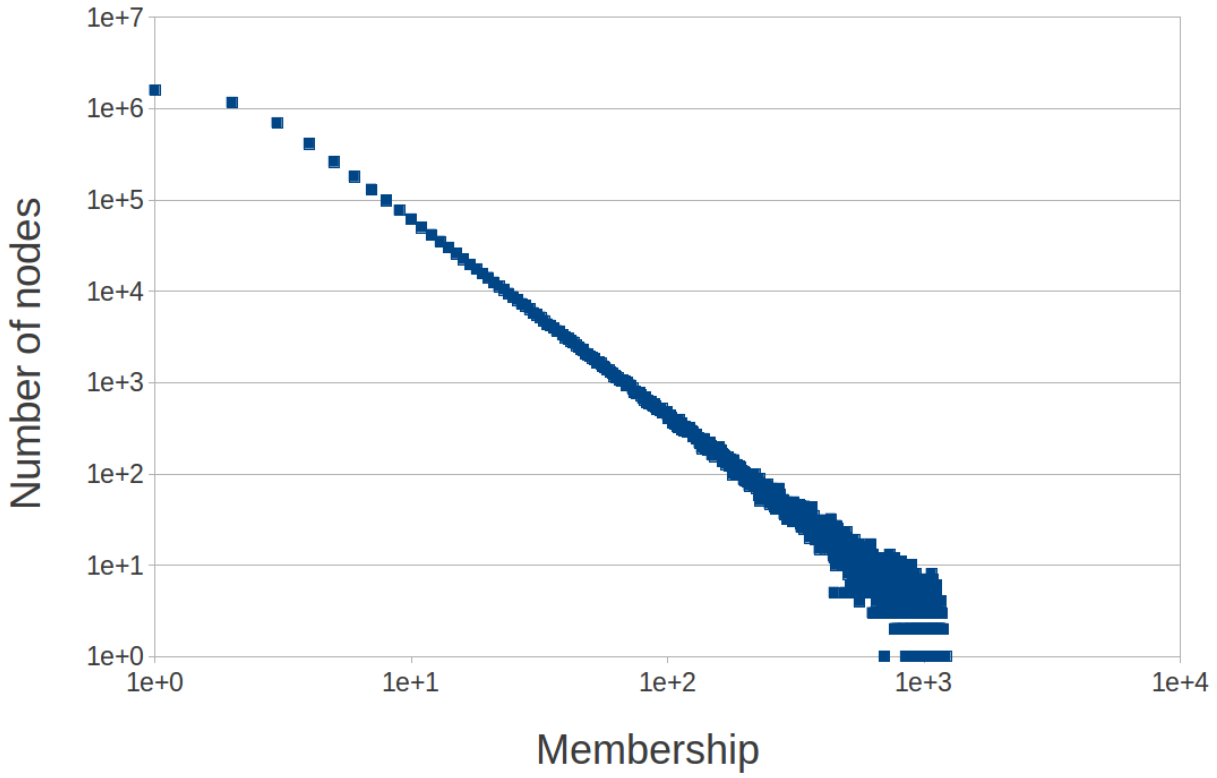


Сравнение LiveJournal vs. СКВ

LiveJournal

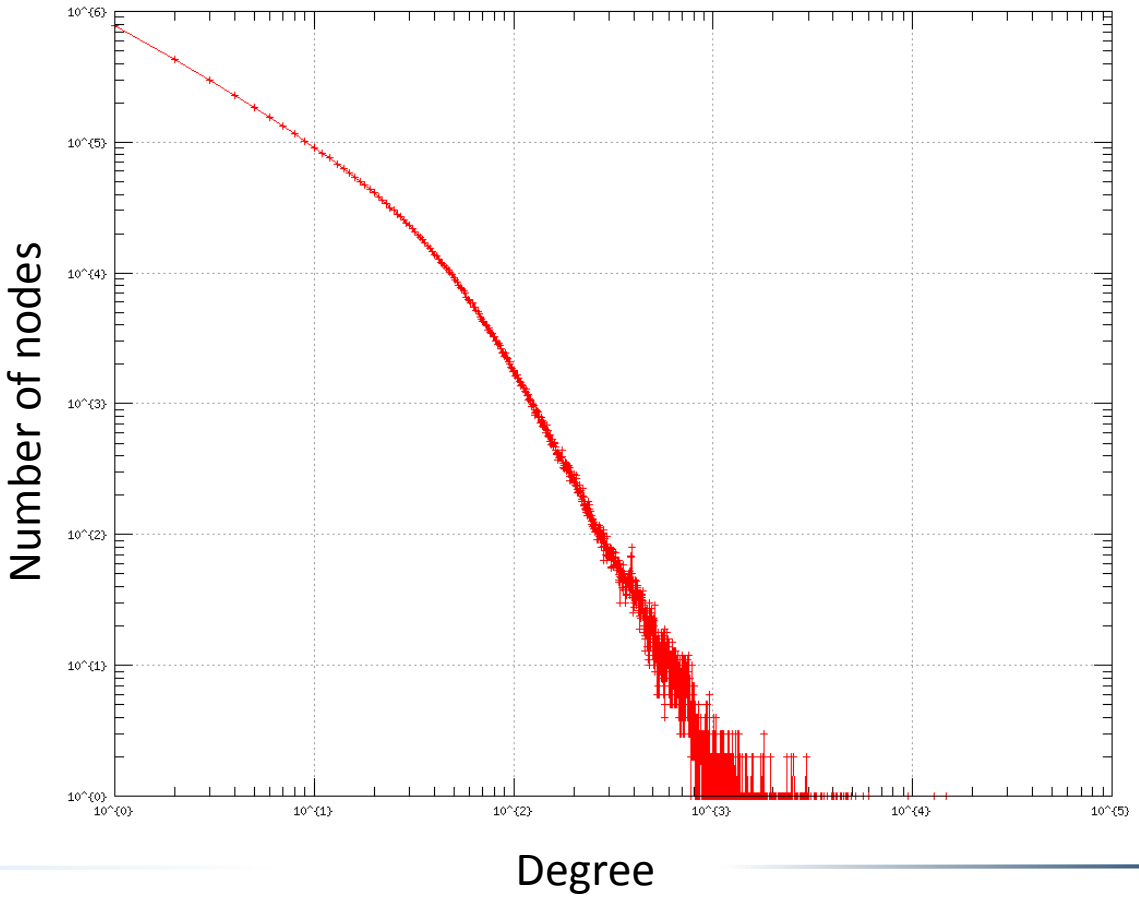


СКВ

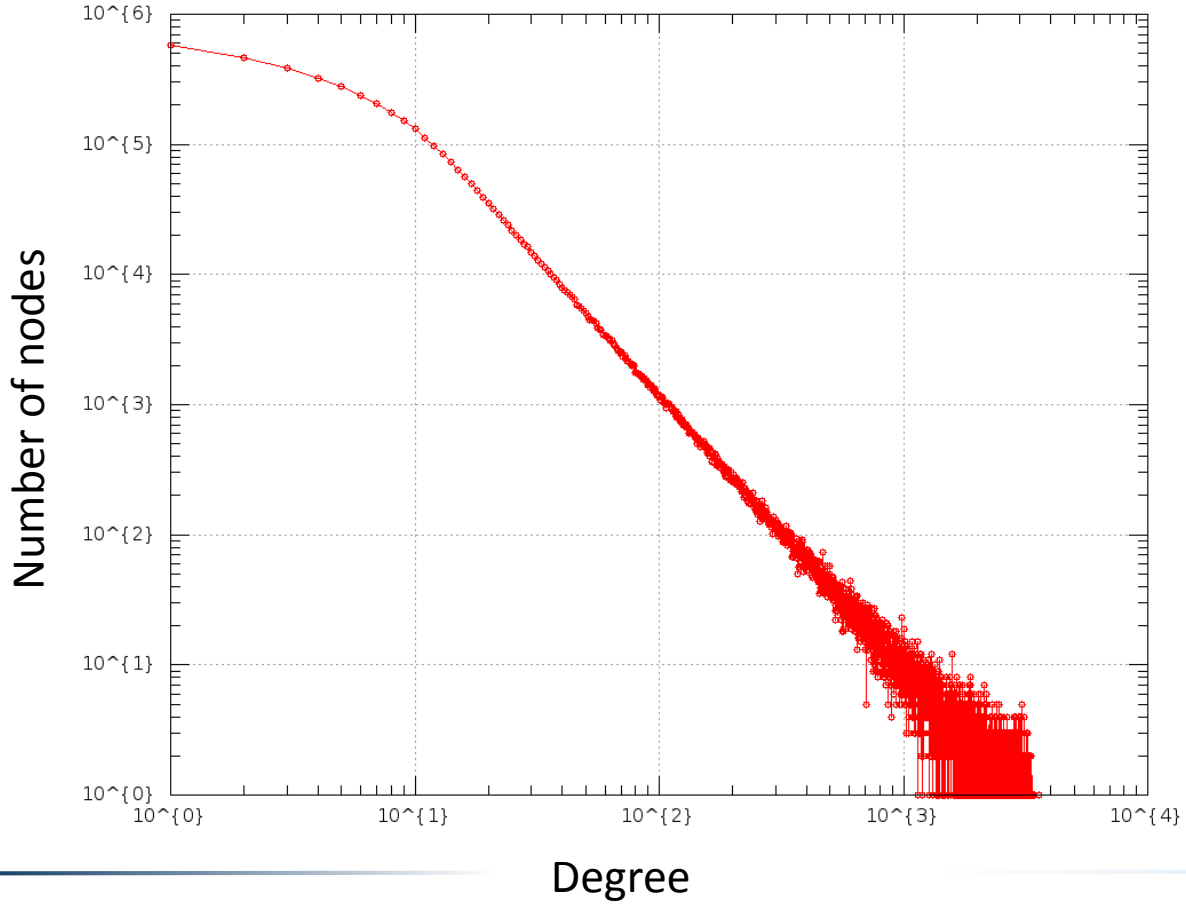


Сравнение LiveJournal vs. СКВ

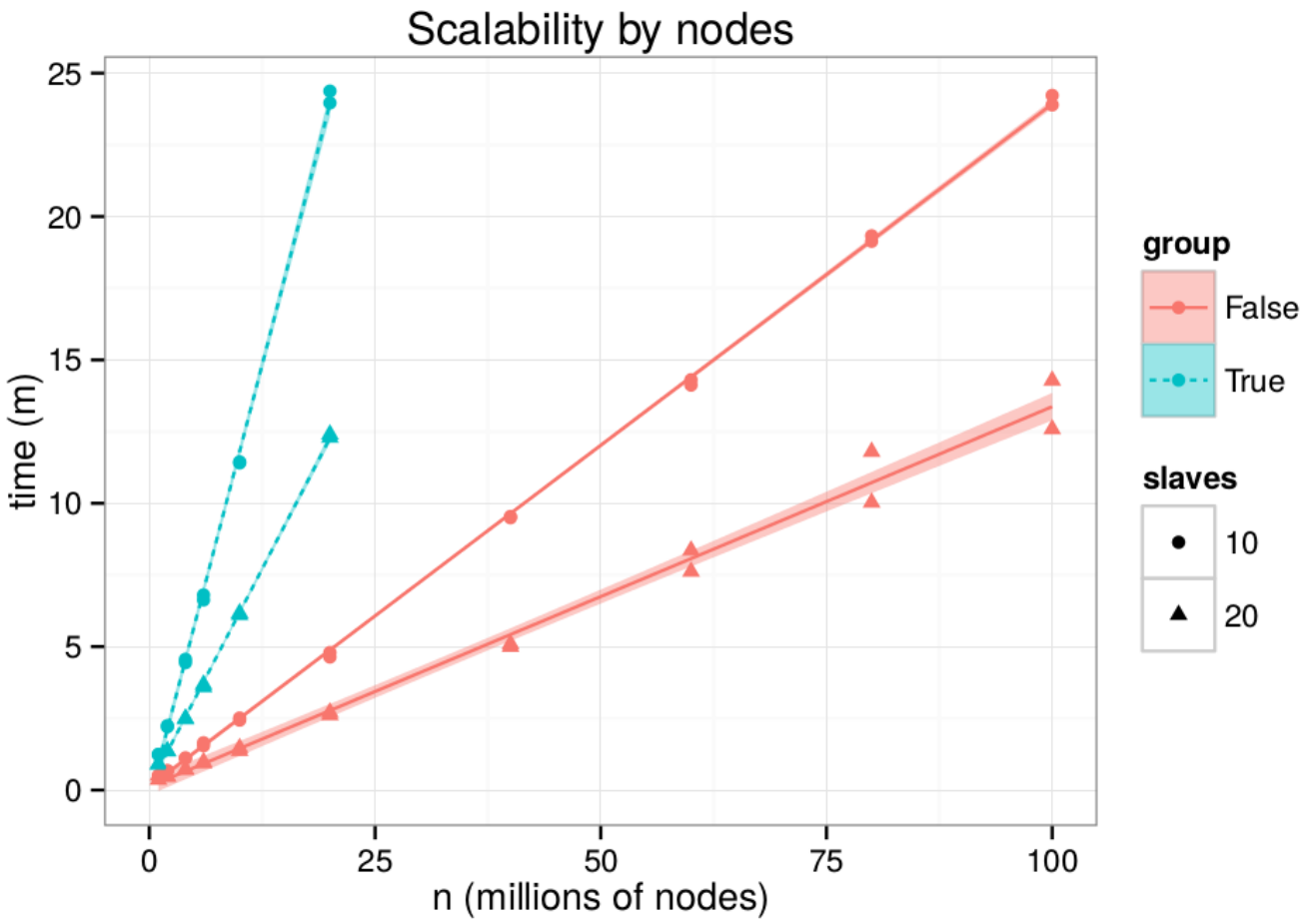
LiveJournal



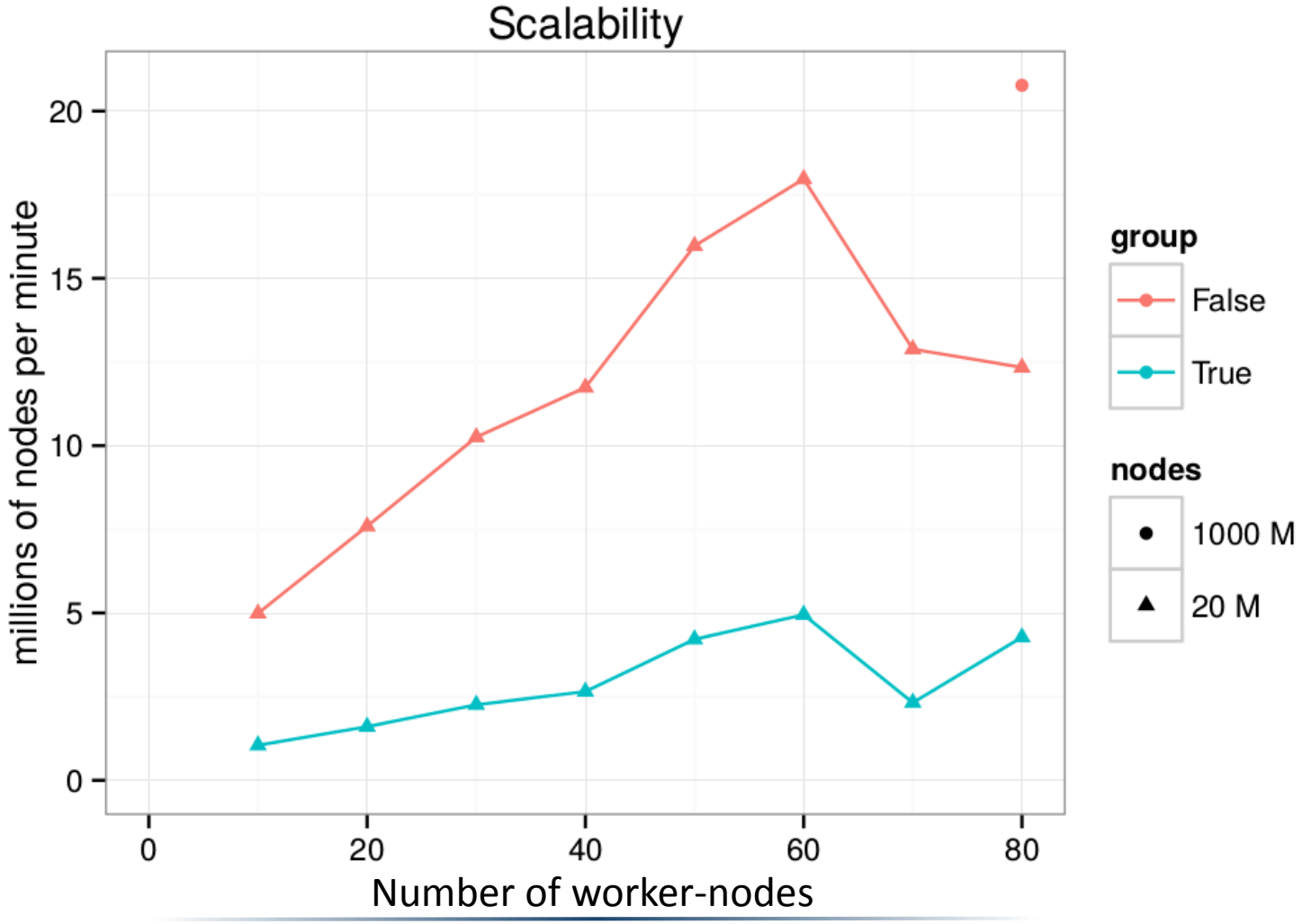
СКВ



Масштабируемость: RGG



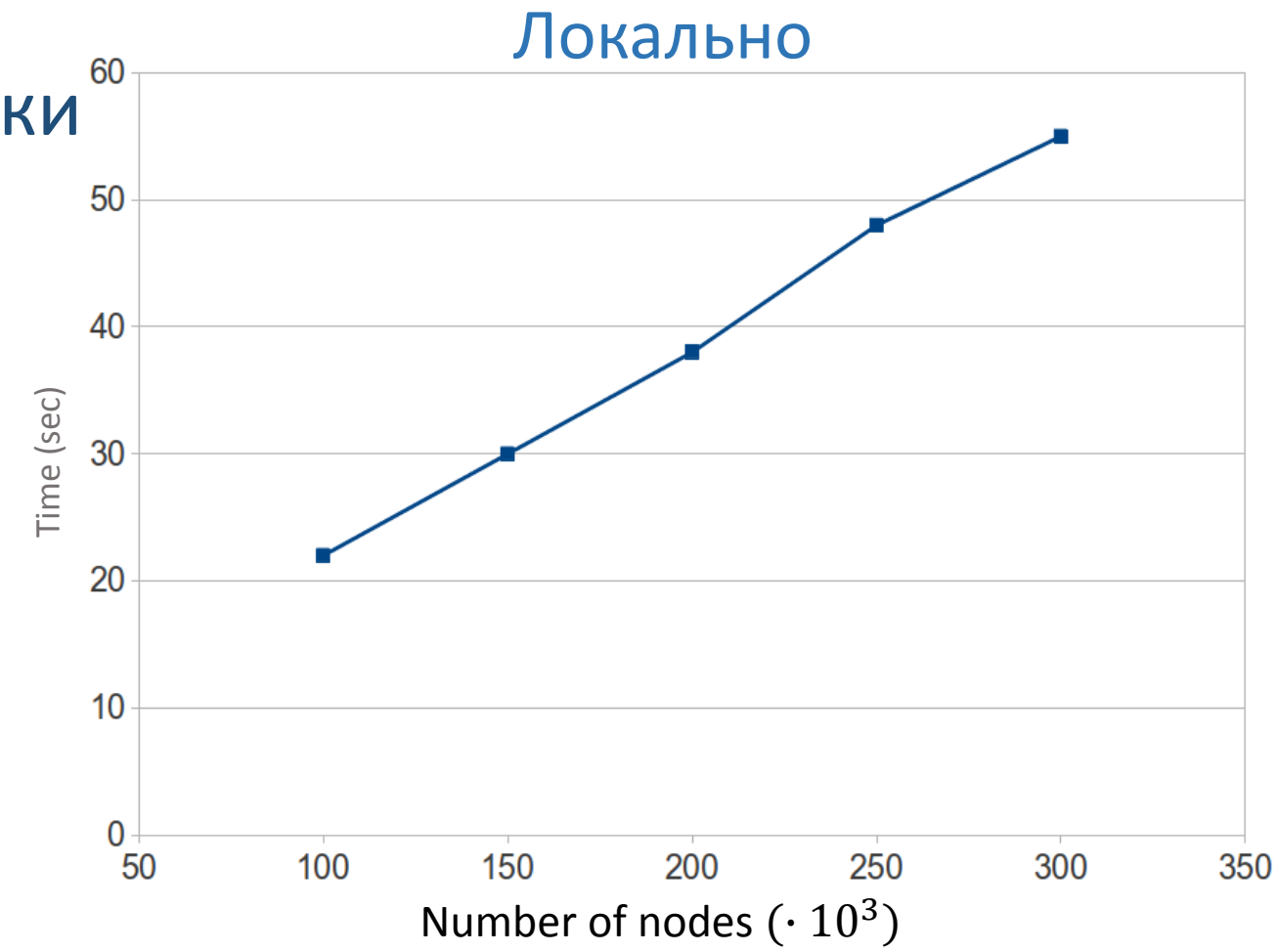
Масштабируемость: RGG



Масштабируемость: СКВ

Параметры генерации для оценки масштабируемости:

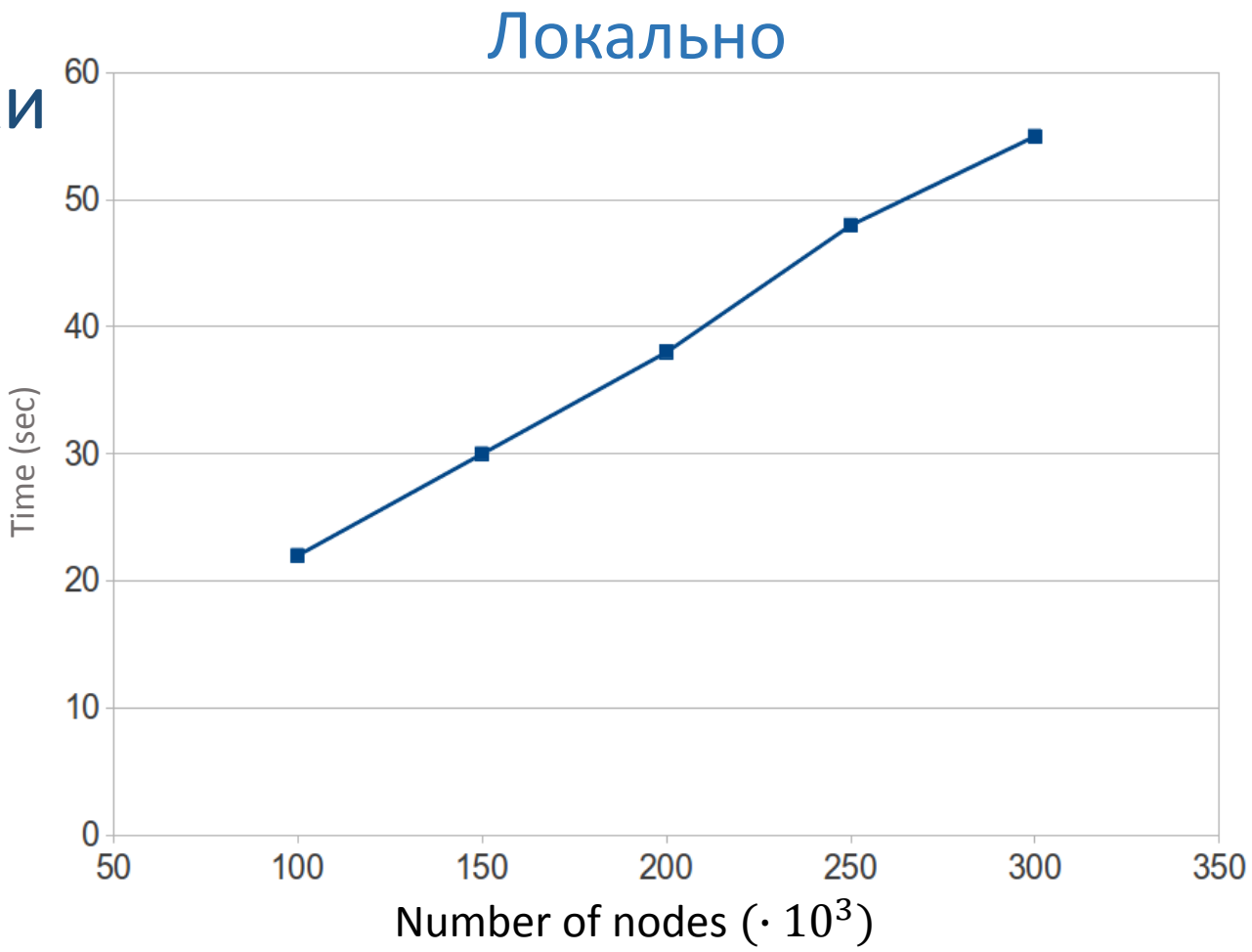
- $\beta_1 = \beta_2 = 2.5$
- $\alpha = 4, \gamma = 0.5$
- $\min\{m_i\} = 1$
- $\min\{c_i\} = 2$
- $\max\{c_i\} = \max\{m_i\} = 10^4$



Масштабируемость: СКВ

Параметры генерации для оценки масштабируемости:

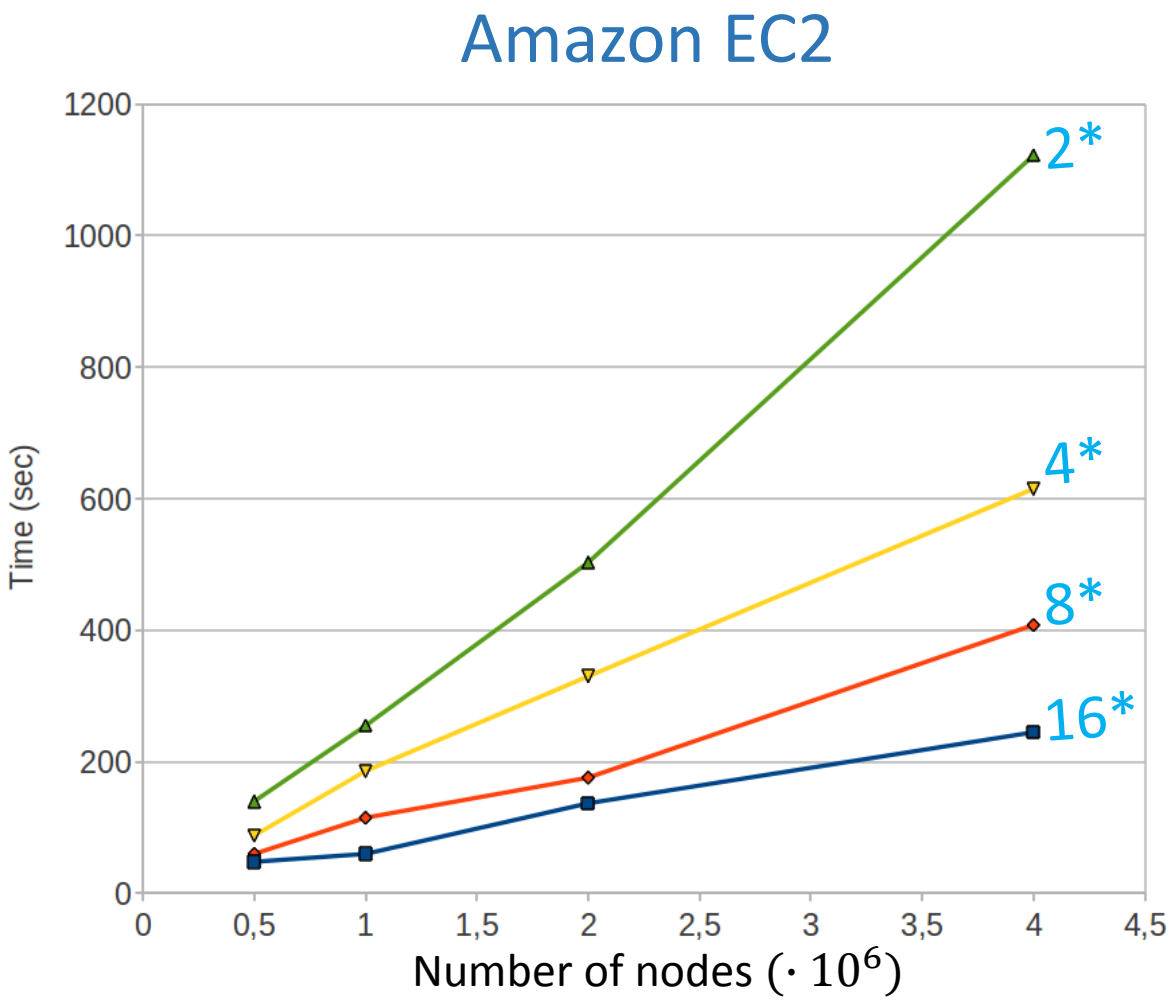
- $\beta_1 = \beta_2 = 2.5$
- $\alpha = 4, \gamma = 0.5$
- $\min\{m_i\} = 1$
- $\min\{c_i\} = 2$
- $\max\{c_i\} = \max\{m_i\} = 10^4$



Масштабируемость: СКВ

Параметры генерации для оценки масштабируемости:

- $\beta_1 = \beta_2 = 2.5$
- $\alpha = 4, \gamma = 0.5$
- $\min\{m_i\} = 1$
- $\min\{c_i\} = 2$
- $\max\{c_i\} = \max\{m_i\} = 10^4$



*Числа на конце линий обозначают количество машин m1.large** в кластере, использовавшимся для генерации.

**m1.large – тип машины на кластере Amazon EC2 (2 vCPU, 7.5 GiB memory, 2x420GB instance storage).

План

- Немного о Spark
- Постановка задачи
- RGG: графы без сообществ
- CKB: графы с сообществами
- Тестирование
- Выводы

Выводы

Описанные инструменты создают реальные сети и имеют ряд преимуществ:

- Графы со свойствами социальных сетей
- Линейная масштабируемость
- Генерация миллиардного графа занимает приемлемое время (55 (150) минут на 80 (150) машинах для RGG (СКВ))
- Гибкость модели

Открытые вопросы

- Тестирование различных алгоритмов поиска сообществ
- Поддержка вариации коэффициента кластеризации
- Взвешенная и иерархическая модификации
- Генерации сообществ пользователей на основе атрибутов и контента

Заключение

Вопросы?

chykhradze@ispras.ru