Pagerank in Apache Flink

Author: Ward Schodts
Supervisor: Juan Soto

Datenbanksysteme und Informationsmanagement
Technische Universität Berlin

July 8, 2016

Agenda

Introduction

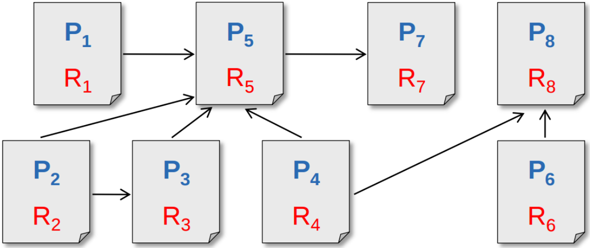
The experiment

The different algorithm implementations

Results

Conclusion

Pagerank



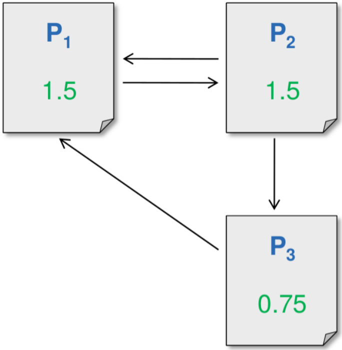
PageRank example 1 [6]

- A page has a high PageRank *R* if
  - there are many pages linking to it
  - or, if there are some pages with a high PageRank linking to it

Pagerank

$$R(P_i) = \sum_{P_j \in B_i} \frac{R(P_j)}{L_j}$$

- where
  - B_i is the set of pages that link to page P_i
  - L_j is the number of outgoing links for page P_j linking to it



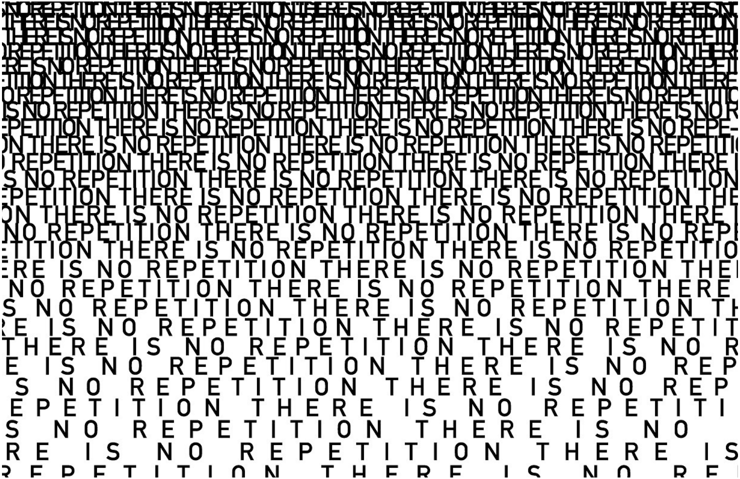
PageRank example 2 [6]

[3]

Apache Flink

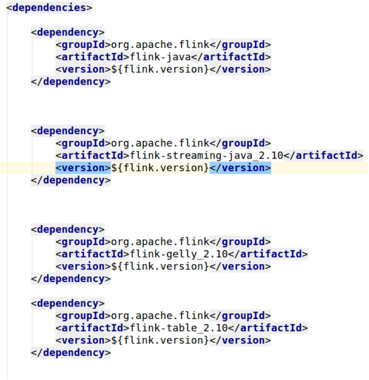
- Open source framework for distributed Big Data Analytics
- Exploits:
  - data streaming
  - in-memory processing
  - iteration operators

  to improve performance
- Formerly Stratosphere (Flink means agile)
- Developped here at TUB



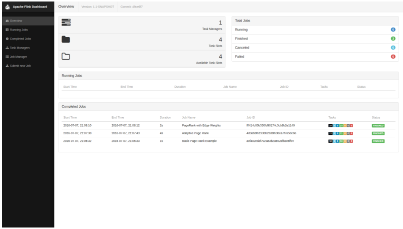
[7]

Apache Flink: 2 possible setups



```
<dependencies>

    <dependency>
        <groupId>org.apache.flink</groupId>
        <artifactId>flink-java</artifactId>
        <version>${flink.version}</version>
    </dependency>


    <dependency>
        <groupId>org.apache.flink</groupId>
        <artifactId>flink-streaming-java_2.10</artifactId>
        <version>${flink.version}</version>
    </dependency>


    <dependency>
        <groupId>org.apache.flink</groupId>
        <artifactId>flink-gelly_2.10</artifactId>
        <version>${flink.version}</version>
    </dependency>

    <dependency>
        <groupId>org.apache.flink</groupId>
        <artifactId>flink-table_2.10</artifactId>
        <version>${flink.version}</version>
    </dependency>
```

Maven



Overview | Version: 1.1-SNAPSHOT | Commit: 49ce6f7

1 Task Managers

4 Task Slots

4 Available Task Slots

Total Jobs	
Running	0
Finished	3
Canceled	0
Failed	0

Running Jobs

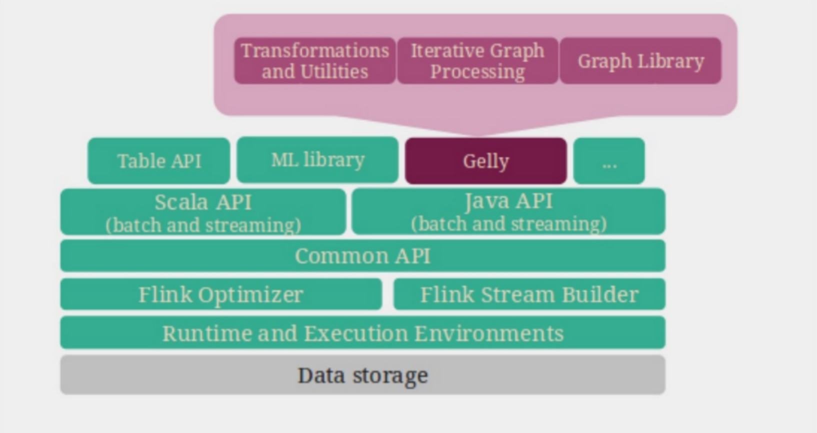
Start Time	End Time	Duration	Job Name	Job ID	Tasks	Status

Completed Jobs

Start Time	End Time	Duration	Job Name	Job ID	Tasks	Status
2016-07-07, 21:08:10	2016-07-07, 21:08:12	2s	PageRank with Edge Weights	ff414c00b530fd90174c3cb8b2e1149	14 0 0 14 0 0 0	FINISHED
2016-07-07, 21:07:38	2016-07-07, 21:07:43	4s	Adaptive Page Rank	4d3ab6f61930b23d6f630ea7f7a50e66	11 0 0 11 0 0 0	FINISHED
2016-07-07, 21:06:32	2016-07-07, 21:06:33	1s	Basic Page Rank Example	acf402ed3f702a83b2a692afb3c6ff97	10 0 0 10 0 0 0	FINISHED

Binary version (self compiled)

Apache Flink: Gelly


Transformations and Utilities
Iterative Graph Processing
Graph Library
Table API
ML library
Gelly
...
Scala API (batch and streaming)
Java API (batch and streaming)
Common API
Flink Optimizer
Flink Stream Builder
Runtime and Execution Environments
Data storage

Gelly

Apache Flink: Gelly

- Large-scale graph processing API
- On top of Flink’s Java API
- Off-the shelf library methods (e.g. pagerank)
- Iterative algorithms

Agenda

Introduction

The experiment

The different algorithm implementations

Results

Conclusion

General experiment setup

Experiment 1:

1. Data file with graph (and pagerank solution)

   ⇓

2. Use Flink and Graphlab implemention to compute pagerank

   ⇓

3. Compare with solution

General experiment setup

Experiment 1:

1. Data file with graph (and pagerank solution)

   ⇓

2. Use Flink and Graphlab implemention to compute pagerank

   ⇓

3. Compare with solution

Experiment 2:

1. Data file with huge graph (no solution yet)

   ⇓

2. Use Flink and Graphlab implemention to compute pagerank

   ⇓

3. Compare with each other

Experiment 1 data

Data from a former Hadoop toolkit (Cloud9, now Bespin):

Name	# vertices	# edges
Small	93	195
Medium	316	430
Large	1458	3545

Experiment 2 data

Webgraph from `snap.stanford.edu/data/`

Name	# vertices	# edges
web-Google	875713	5105039

Agenda

Introduction

The experiment

The different algorithm implementations

Results

Conclusion

Flink algorithm 1

dataArtisians logo, [1]

- An exercise from dataArtisians
- Uses the standard Gelly implementation
- # input nodes = # output nodes

Flink algorithm 2

dataArtisans

- A case study implementation from dataArtisians
- A custom implementation
- # input nodes = # output nodes

Flink algorithm 3

- An example from the Apache Flink repository
- A custom implementation
- # input nodes != # output nodes → filters

Turi pagerank algorithm

Turi logo, [8]

- Used the standard implementation
- Builds a graph out of the edges dataset

Agenda

Introduction

The experiment

The different algorithm implementations

Results

Conclusion

Results

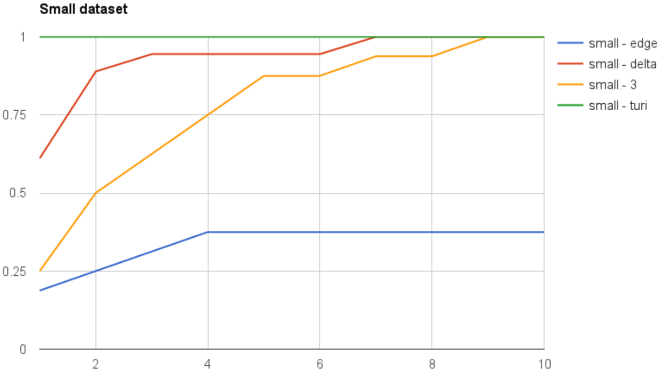
To compare I implemented a comparer that:

- could handle list of diffrence sizes,
- took care of equal pagerank values (they maybe sorted in different way),
- had a modifyable window to compare with.

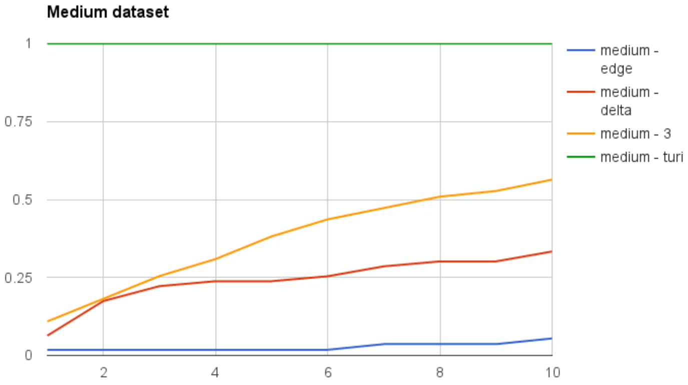
Results: experiment 1

Any expectations?

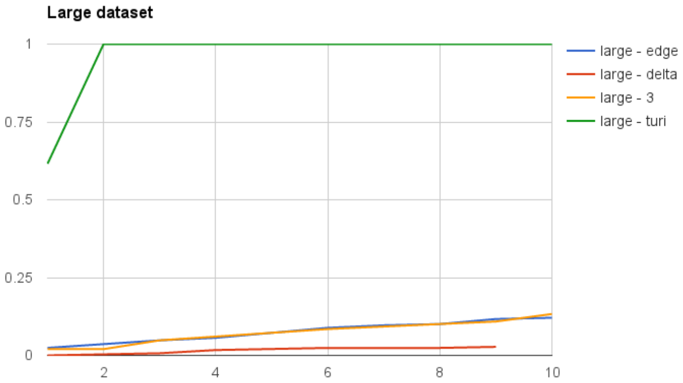
Results: experiment 1



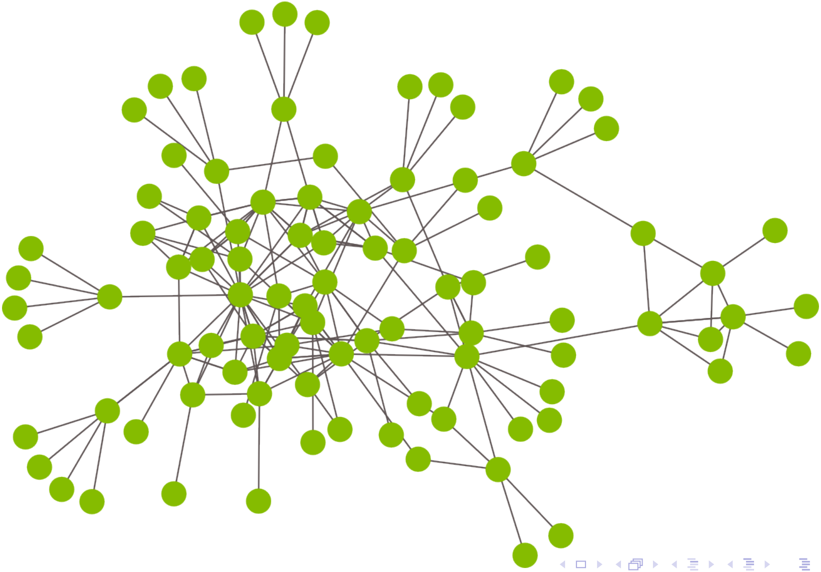
Results: experiment 1



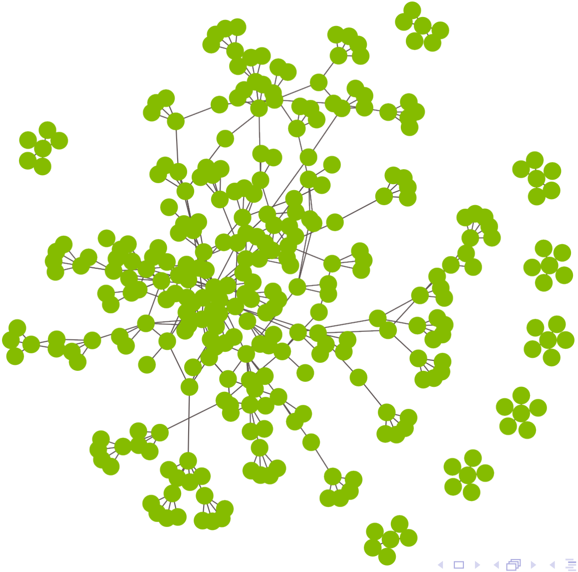
Results: experiment 1



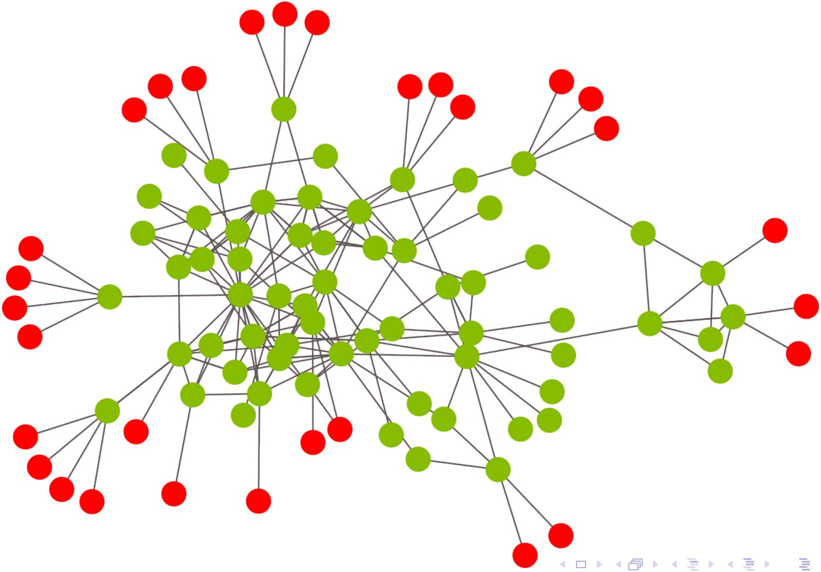
Why are the results so bad?



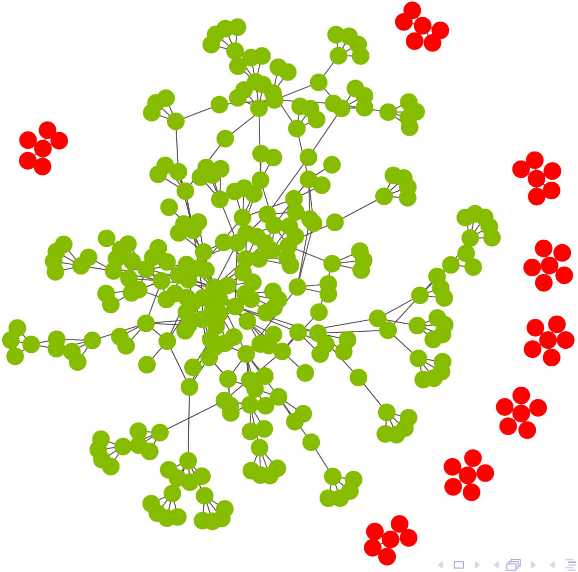
Why are the results so bad?



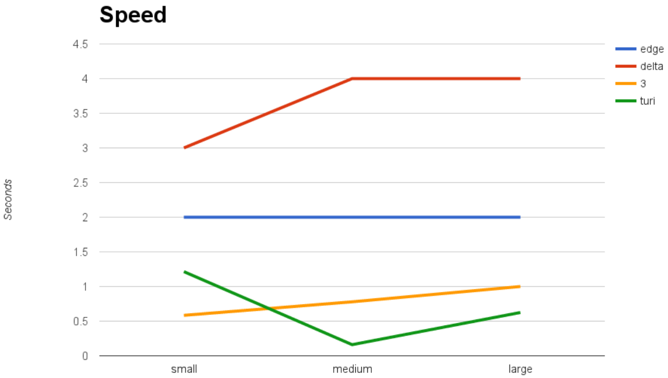
Why are the results so bad?



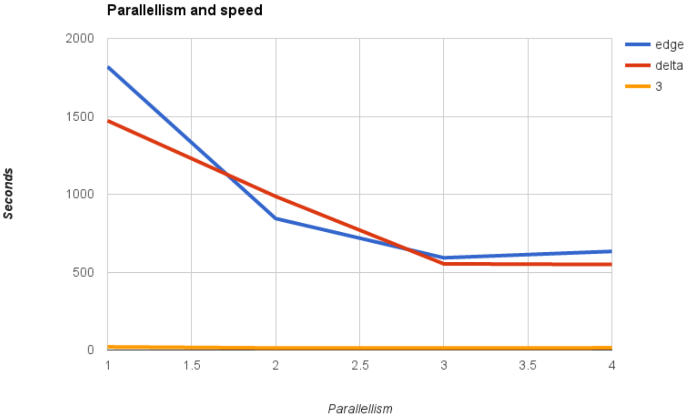
Why are the results so bad?



Speed of experiment 1



Speed of experiment 2



Agenda

Introduction

The experiment

The different algorithm implementations

Results

Conclusion

Conclusion

Thank you for your attention

Questions?

References I

- Data Artisians. *Data Artisians logo*. URL: %5Curl%7Bhttps://www.mapr.com/sites/default/files/data_artisans_logo.png%7D.
- Slim Baltagi. *Overview of Apache Flink: Next-Gen Big Data Analytics Framework*. 2015. URL: %5Curl%7Bhttp://www.slideshare.net/sbaltagi/overview-of-apacheflinkbyslimbaltagi?qid=5f0b5424-d187-4c79-a600-6cae794c686e&v=&b=&from_search=3%7D.
- Apache Flink. *Apache Flink Squirrel*. URL: %5Curl%7Bhttps://flink.apache.org/img/logo/png/1000/flink_squirrel_1000.png%7D.

References II

- Lawrence Page et al. "The PageRank citation ranking: bringing order to the web." In: (1999).

- Lawrence Page et al. "The PageRank citation ranking: bringing order to the web." In: (1999).

- Beat signer. *Google PageRank*. 2009. URL: %5Curl%7Bhttp://www.slideshare.net/signer/google-pagerank-presentation?qid=18af8836-30e7-41cd-9edb-956bd7ca324d&v=&b=&from_search=2%7D.

References III

- Mathias Spahlinger. *There is no repetition*. URL: %5Curl%7Bhttps://www.google.com/search?q=repeat&source=lnms&tbm=isch&sa=X&ved=0ahUKEwi4laH2tuLNAhVnB8AKHTPQCU4Q_AUICCgB&biw=1590&bih=765#tbm=isch&q=no+repetition&imgrc=h1qwLbEEezv8SM:%7D.
- Inc Turi. *Turi*. URL: %5Curl%7Bhttps://www.google.com/search?q=repeat&source=lnms&tbm=isch&sa=X&ved=0ahUKEwi4laH2tuLNAhVnB8AKHTPQCU4Q_AUICCgB&biw=1590&bih=765#tbm=isch&q=no+repetition&imgrc=h1qwLbEEezv8SM:%7D.