

# Pagerank in Apache Flink

Author: Ward Schodts

Supervisor: Juan Soto

Datenbanksysteme und Informationsmanagement  
Technische Universität Berlin



July 8, 2016

# Agenda



Introduction

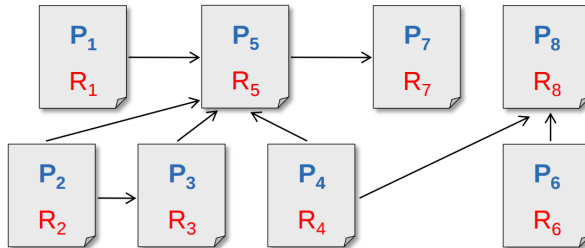
The experiment

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

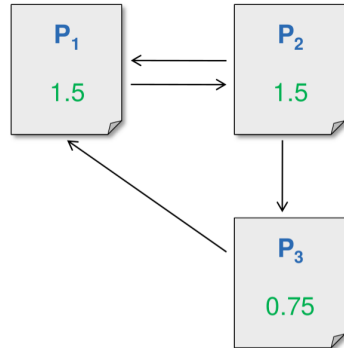
# Pagerank



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

► 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 operatorsto improve performance
- ▶ Formerly Stratosphere (Flink means agile)
- ▶ Developed here at TUB





# 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>
</dependencies>
```

Maven

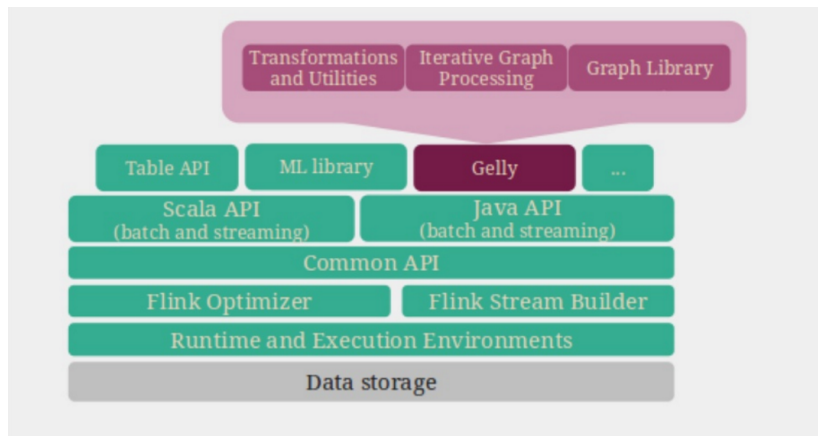
The screenshot shows the Apache Flink dashboard with a sidebar on the left containing navigation links: Overview, Jobs, Clusters, and Configurations. The main content area displays a table of jobs. The 'Running Jobs' section shows a single job in a 'Running' state. Below this, the 'Completed Jobs' section shows a list of jobs with their status, start time, end time, and other details. The jobs listed are 'flink-java', 'flink-streaming-java', 'flink-gelly', and 'flink-table', all of which are in a 'Completed' state.

Job Name	Start Time	End Time	Status	Job ID	Job Type	Job State
flink-java	2020-07-07 12:00:00	2020-07-07 12:00:00	Completed	flink-java	Job	Completed
flink-streaming-java	2020-07-07 12:00:00	2020-07-07 12:00:00	Completed	flink-streaming-java	Job	Completed
flink-gelly	2020-07-07 12:00:00	2020-07-07 12:00:00	Completed	flink-gelly	Job	Completed
flink-table	2020-07-07 12:00:00	2020-07-07 12:00:00	Completed	flink-table	Job	Completed

Binary version (self compiled)



# Apache Flink: Gelly



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 implementation 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 implementation to compute pagerank



3. Compare with solution

## Experiment 2:

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



2. Use Flink and Graphlab implementation to compute pagerank



3. Compare with each other

# Experiment 1 data



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

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

# Experiment 2 data



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

Name	# vertices	# edges
web-Stanford	281903	2312497
web-Google	875713	5105039

# Agenda



Introduction

The experiment

The different algorithm implementations

Results

Conclusion



# Flink algorithm 1



**dataArtisans**

Turi, [8]

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

# Flink algorithm 2



**dataArtisans**

Turi, [8]

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

# Flink algorithm 3



Turi, [8]

- ▶ An example from the Apache Flink repository
- ▶ A custom implementation
- ▶ # input nodes  $\neq$  # output nodes  $\rightarrow$  filters

# Turi pagerank algorithm



Turi, [8]

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

# Agenda



Introduction

The experiment

The different algorithm implementations

**Results**

Conclusion

# Results



# Agenda



Introduction

The experiment

The different algorithm implementations

Results

Conclusion

# Conclusion





# Thank you for your attention



## Questions?



# References I



**Data Artisans.** *Data Artisans 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.