

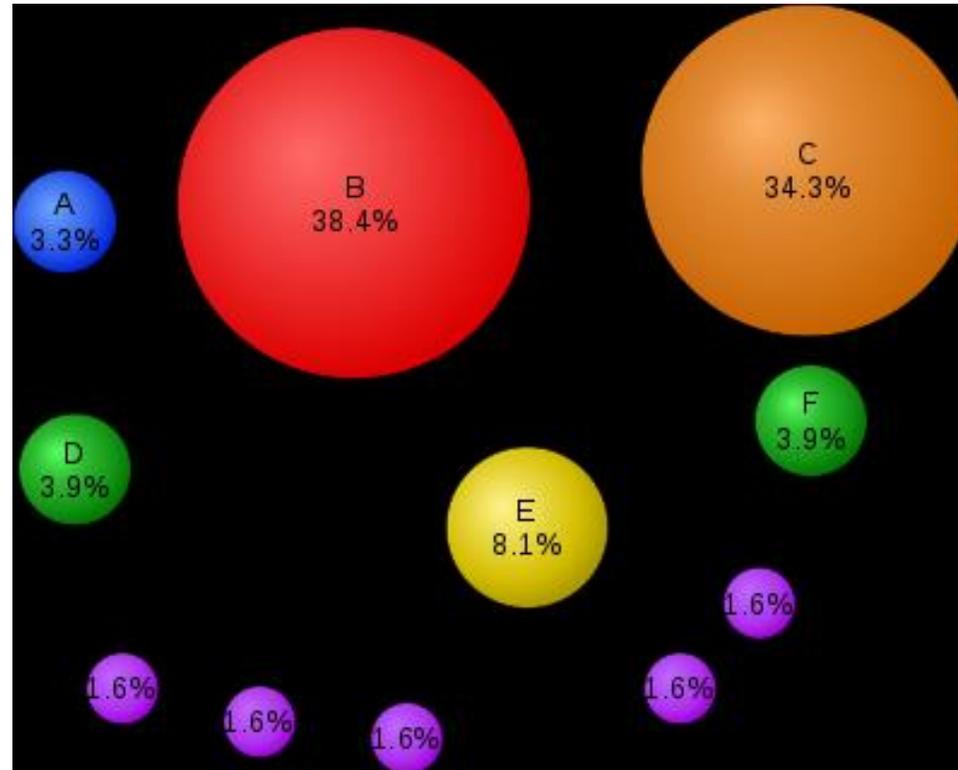
The background features abstract, overlapping green geometric shapes in various shades, creating a modern and dynamic feel. The shapes are primarily triangles and polygons, some with thin white outlines, set against a white background.

PageRank Analysis, Implementation & Optimization

Chaoran Chen

Outline

- ▶ Motivation
- ▶ Review
- ▶ Implementation
- ▶ Visualization
- ▶ Experiment results on large datasets
- ▶ Conclusion
- ▶ References



Motivation

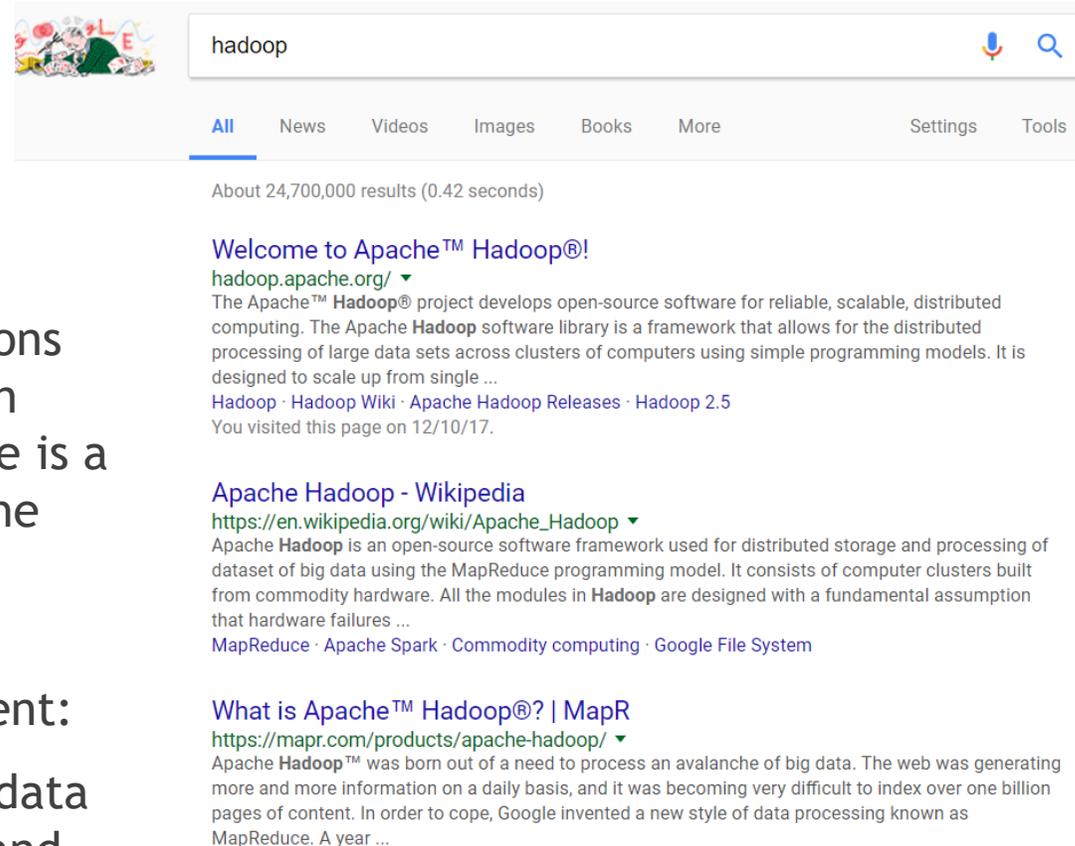
- ▶ Need for PageRank:

The Search engines store billions of web pages which overall contain trillions of web URL links. So, there is a need for an algorithm that gives the most relevant pages specific to a query.

- ▶ Need for Distributed Environment:

Trillions of links implies huge data storage required. So Map-Reduce and Distributed Storage is needed.

- ▶ How to improve?



The image shows a screenshot of a Google search for the term "hadoop". The search bar at the top contains the word "hadoop" and has a search icon to its right. Below the search bar, there are tabs for "All", "News", "Videos", "Images", "Books", and "More", with "All" selected. The search results are displayed below, showing "About 24,700,000 results (0.42 seconds)". The first result is "Welcome to Apache™ Hadoop@!" with the URL "hadoop.apache.org/". The second result is "Apache Hadoop - Wikipedia" with the URL "https://en.wikipedia.org/wiki/Apache_Hadoop". The third result is "What is Apache™ Hadoop@? | MapR" with the URL "https://mapr.com/products/apache-hadoop/".

hadoop

All News Videos Images Books More Settings Tools

About 24,700,000 results (0.42 seconds)

[Welcome to Apache™ Hadoop@!](#)
hadoop.apache.org/
The Apache™ **Hadoop**® project develops open-source software for reliable, scalable, distributed computing. The Apache **Hadoop** software library is a framework that allows for the distributed processing of large data sets across clusters of computers using simple programming models. It is designed to scale up from single ...
[Hadoop](#) · [Hadoop Wiki](#) · [Apache Hadoop Releases](#) · [Hadoop 2.5](#)
You visited this page on 12/10/17.

[Apache Hadoop - Wikipedia](#)
https://en.wikipedia.org/wiki/Apache_Hadoop
Apache **Hadoop** is an open-source software framework used for distributed storage and processing of dataset of big data using the MapReduce programming model. It consists of computer clusters built from commodity hardware. All the modules in **Hadoop** are designed with a fundamental assumption that hardware failures ...
[MapReduce](#) · [Apache Spark](#) · [Commodity computing](#) · [Google File System](#)

[What is Apache™ Hadoop@? | MapR](#)
<https://mapr.com/products/apache-hadoop/>
Apache **Hadoop**™ was born out of a need to process an avalanche of big data. The web was generating more and more information on a daily basis, and it was becoming very difficult to index over one billion pages of content. In order to cope, Google invented a new style of data processing known as MapReduce. A year ...

Review

- ▶ PageRank x_p of p is computed by taking into account the set of pages $pa[p]$ pointing to p .

$$x_p = d \sum_{q \in pa[p]} \frac{x_q}{h_q} + (1 - d) .$$

Here h_q is the outdegree of q , that is the number of hyperlinks outcoming from q . Let d be a factor used for normalization so that the total rank of all web pages is constant.

- ▶ When stacking all the x_p into vector x , we get

$$x = dWx + (1 - d)I .$$

Here $W = \{w_{i,j}\}$ is transition matrix. $w_{i,j} = 1/h_j$ if there is a hyperlink from j to i and $w_{i,j} = 0$ otherwise.

Review

- ▶ Stochastic Interpretation:

$$x(t + 1) = dWx(t) + (1 - d)I_N$$

PageRank dynamic system(random walk), stable after n iterations, proved by Markov Chain Theory.

- ▶ Dumping factor d:

If $d = 0$, all the PageRanks equals 1.

If $d = 1$, many pages would have a zero PageRank.

- ▶ Dangling pages: pages w/o hyperlinks

Handling: introducing a dummy node or removing dangling pages.

Review

► Communities and Energy Balance

A community could be a set of pages on a given topic, the researchers' home pages or a Website; the corresponding energy is a measure of its authority.

$$E_I = |I| + E_I^{in} - E_I^{out} - E_I^{dp}$$

$|I|$: # of pages, “default energy”

E_I^{in} : Page Rank inside the community, communities with many references have a high authority

E_I^{out} : Page Rank outside the community , having hyperlinks outside the community leads to decrease energy

E_I^{dp} : the presence of pages without hyperlinks yields a loss of energy

Review

Energy Calculation, determined by d , W and x

THEOREM 4.2. *Given a community G_I , let f_p be the fraction of the hyperlinks of page p that point to pages in G_I with respect to the total number of hyperlinks outgoing from p . Let E_I^{in} , E_I^{out} , and E_I^{dp} be defined by*

$$E_I^{in} = \frac{d}{1-d} \sum_{i \in \text{in}(I)} f_i x_i^*, \quad E_I^{out} = \frac{d}{1-d} \sum_{i \in \text{out}(I)} (1 - f_i) x_i^*, \quad E_I^{dp} = \frac{d}{1-d} \sum_{i \in \text{dp}(I)} x_i^*.$$

Then, PageRank x_i^ of G_I satisfies*

$$E_I = |I| - E_I^{dp} + E_I^{in} - E_I^{out}. \quad (23)$$

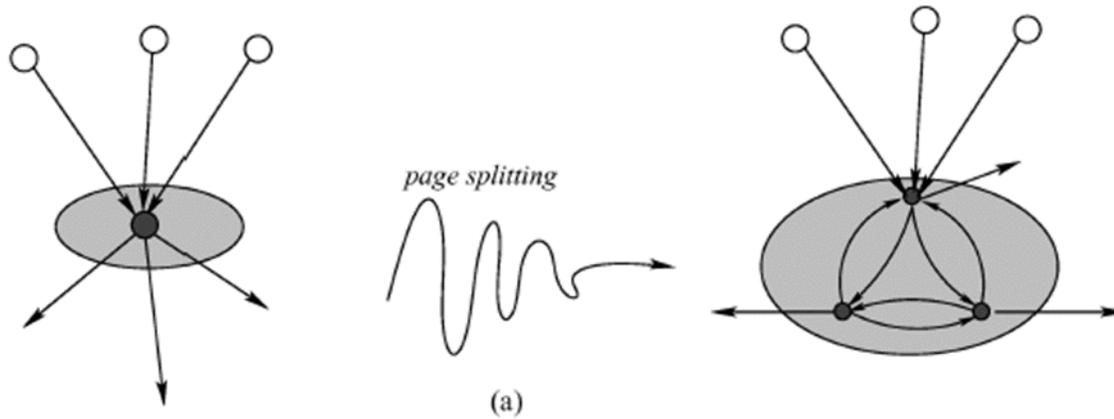
$$\text{Energy Loss} = E_I^{out} + E_I^{dp}$$

Review

► Page Promotion:

Splitting into multiple pages.

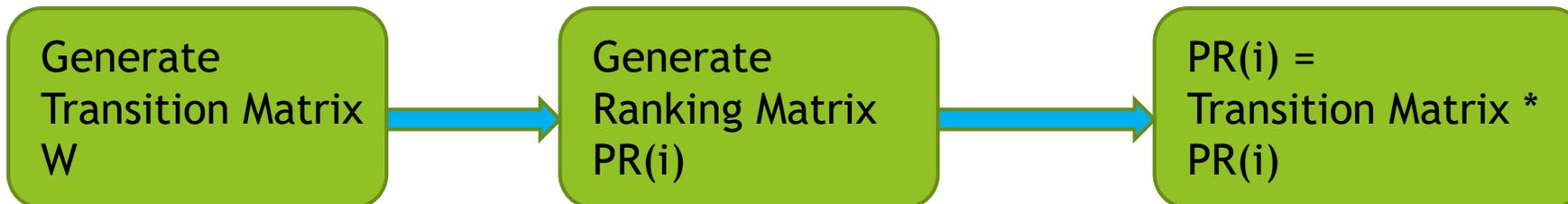
The same content divided into many small pages yields a higher score than the same content into a single large page.
Increase the PageRanks.



Implementation

- ▶ Convert each URL into a unique integer.
- ▶ Store each hyperlink in a database using the integer IDs to identify pages.
- ▶ $PR(n) = \text{Transition Matrix} * PR(n-1)$

```
1 http://www1.hollins.edu/  
2 http://www.hollins.edu/  
3 http://www1.hollins.edu/Docs/CompTech/Network/webmail_faq.htm  
4 http://www1.hollins.edu/Docs/Forms/GetForms.htm  
5 http://www1.hollins.edu/Docs/misc/travel.htm  
6 http://www1.hollins.edu/Docs/GVCalendar/gvmain.htm  
7 http://www1.hollins.edu/docs/events/events.htm
```



Implementation

► Dataset:

Wiki-Vote, Nodes: 7115, Edges: 103689

soc-Epinions, Nodes: 75879, Edges: 508837

► Input format:

FromNodeIdToNodeId

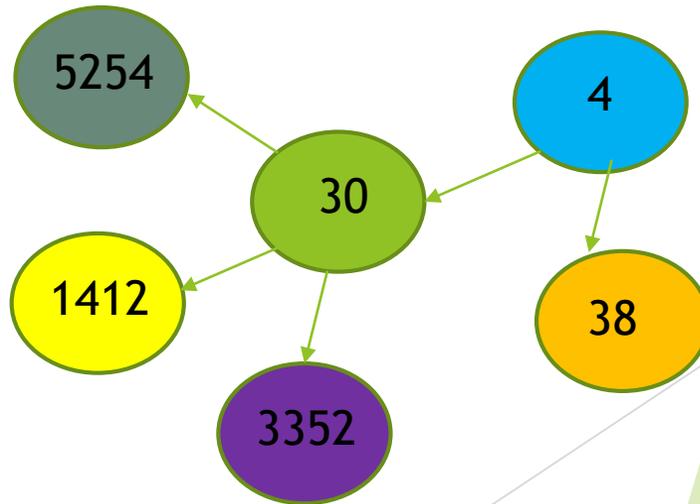
30	1412
30	3352
30	5254
4	30
4	38

W	30	38	1412	3352	5254
4	1/2	1/2	0	0	0
30	0	0	1/3	1/3	1/3



 **Wikipedia vote network**

 **Epinions social network**



How to calculate PR1?

→ PR1 = Transition Matrix * PR0

To\From	WA	WB	WC	WD
WA	0	1/2	1	0
WB	1/3	0	0	1/2
WC	1/3	0	0	1/2
WD	1/3	1/2	0	0

transition.txt(dataset)



	PR0
A	1/4
B	1/4
C	1/4
D	1/4

PR0.txt
initial vector

	PR1
A	9/24
B	5/24
C	5/24
D	5/24

PR1.txt

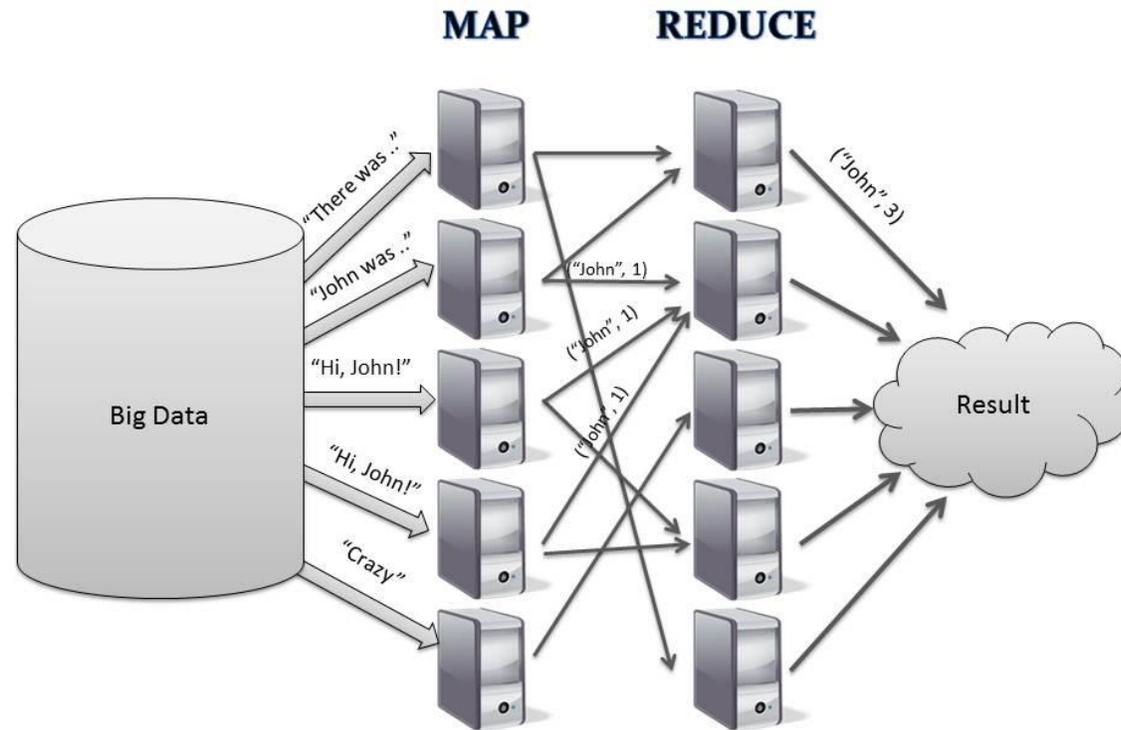
Implementation



- ▶ Apache Hadoop MapReduce
- ▶ Preprocessing:

Client read input from the file that contains the dataset and build the transition matrix W .

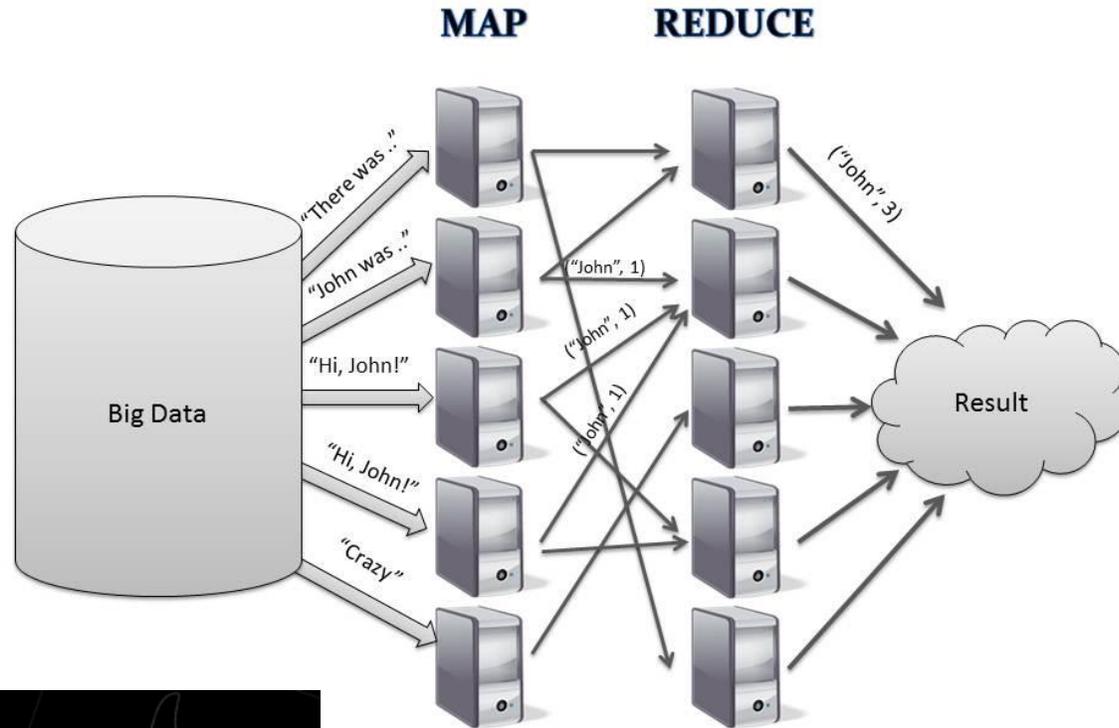
Client read input from initial PageRank file PR0.txt to build the PR vector x .



Implementation



- ▶ Mapper: leverage the job onto multiple machines.
- ▶ Reducer: compute the ranking value on different machines and combine the results into a single final result.
- ▶ The input and output of Mapper and Reducer are $\langle \text{Key}, \text{Value} \rangle$ pairs, which can be stored in HBase.



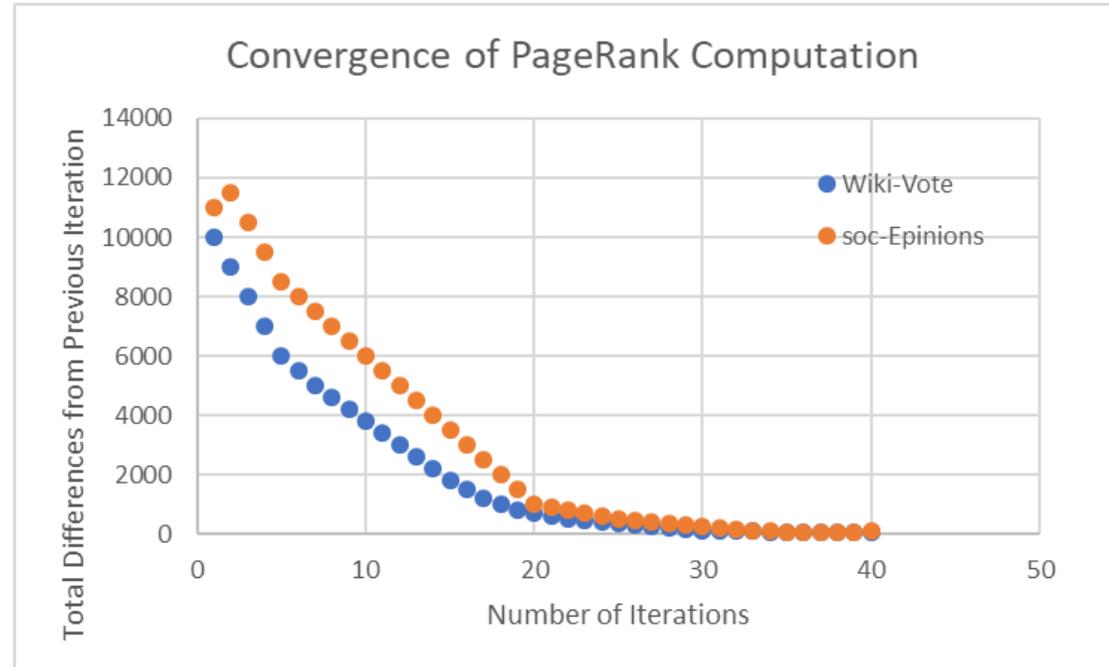
Visualization

- ▶ http://localhost/pagerank_search/index.html?query=756



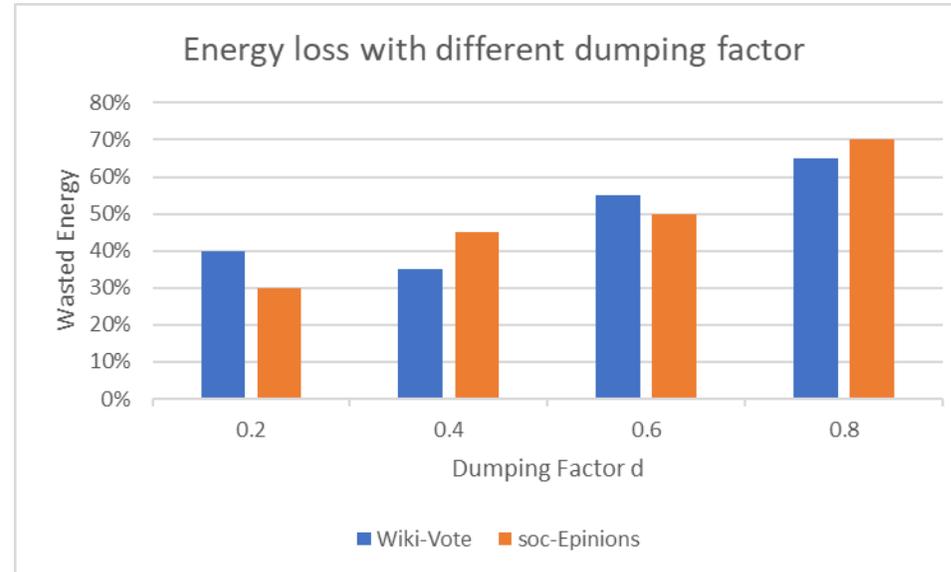
Experiment Results

- ▶ Stability
- ▶ The result will converge after certain number of iterations.



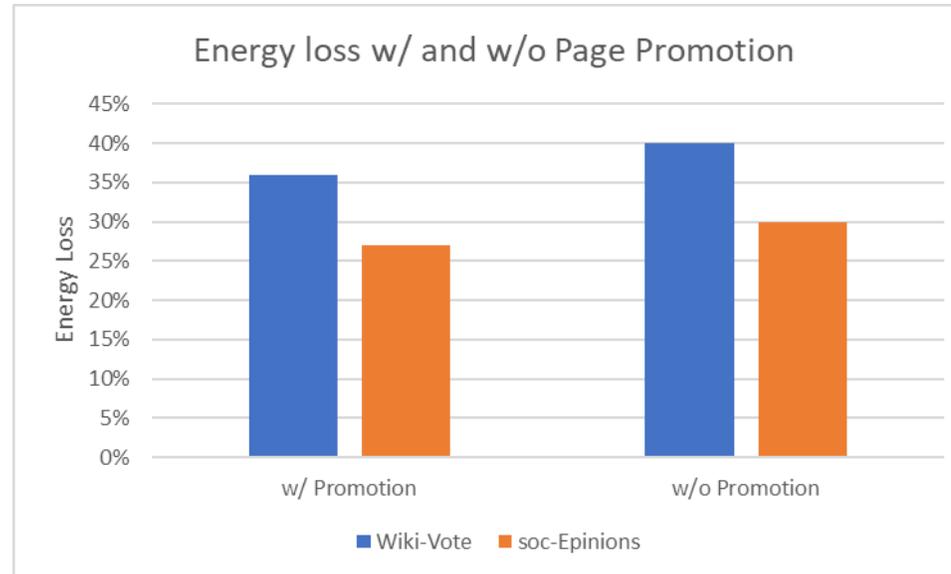
Experiment Results

- ▶ Different dumping factor
- ▶ When d approaches to 1(0.8), the loss can be an important percentage of the available energy
- ▶ Better performance when d is between 0.2 and 0.6



Experiment Results

- ▶ Optimization: Page Promotion
- ▶ Tried Splitting the content of pages to promote the performance.



Conclusion

- ▶ In order to maximize the efficiency of the search engines, we need to reduce the energy loss of the system as much as possible.
- ▶ PageRank is strongly affected by the choice of the dumping factor d . If d approaches 1, the loss can be an important percentage of the available energy.
- ▶ Page Promotion is an effective way to promote the overall PageRank of the entire page community.
- ▶ Future work: Trying different Page Promotion strategies in more large datasets and compare the performance. Build a real-world web ranking application for ranking different networks.

References

- ▶ Page, L., Brin, S., Motwani, R., and Winograd, T.
The PageRank citation ranking: Bringing order to the web.
Stanford InfoLab, 1998.
- ▶ MONICA BIANCHINI, MARCO GORI, and FRANCO SCARSELLI
Inside PageRank.
University of Siena, 2000.
- ▶ Surajkumar Harikumar and Manikandan Srinivasan
Google's PageRank Algorithm : An Analysis, Implementation and Relevance today.
- ▶ <http://www.ams.org/samplings/feature-column/fcarc-pagerank>

Thanks!

