J-KOMA Journal of Computer Science and Applications

Universitas Negeri Jakarta
Phone: (021) 4890856
East Jakarta City
Jakarta, 13220

Open Access

E-ISSN: *2620 - 4872*
Vol.*07*, ***No***.*02*
Doi: *https://doi.org/10.21009/j-koma.v7i2.01*
Recivied: *03 Jul 2024*
Accepted: *10 Agst 2024*
Published: *20 Des 2024*

Keywords:
PageRank Algorithm;
Modified DPC;
Random Walker;

Correspondence Email:
farhan.herdia123@gmail.com

Comparison of PageRank Algorithm Implementations on a Single Computer

Farhan Herdian Pradana[1]*, Muhammad Eka Suryana[2], Med Irzal[3], Ersa Resita[4]

[1,2,3]Computer Science Study Program, Universitas Jakarta

Abstract (10pt)
Pagerank Algorithm is an algorithm used for calculating web page ranking in Google search engine. Problem arises for Pagerank Algorithm due to big main memory usage, thus make it impossible to run in single machine computer with limited main memory. Alternative algorithms will be proposed by comparing the alternative algorithms from other studies with the Original Google Pagerank in terms of speed, main memory usage, and their result similarity. In this study, the Orignal Pagerank, Distributed Pagerank Computation (DPC), Modified DPC, and Random Walker algorithms will be implemented. The implemented algorithms will be run with datasets, and their speed, main memory usage, and result similarity will be compared. For result similarity, Random Walker's result will be used as a benchmark, since it has been used as base concept of Pagerank. It is concluded that the Original Pagerank is faster and has very similar result with Random Walker, while DPC and MDPC have significantly smaller main memory usage, thus very suitable for single machine computer with limited main memory, but run slower and sacrificing result similarity.

INTRODUCTION

The internet is a vast network that connects computers worldwide, operated by companies, governments, universities, and other organizations, to communicate with each other (Sample, 2018). The internet has many uses. In the field of communications, the internet gave rise to Voice over Internet Protocol (VoIP) and electronic mail (email). In the field of data transmission, the internet allows users to upload files to file servers to share with others or access them anywhere. Most popularly, aside from these two areas, the internet gave rise to the World Wide Web (WWW), which allows websites, commonly referred to as websites, to be accessible to everyone.A website is a collection of interconnected web pages under the same domain name. Websites can be created and maintained by an individual, group, company, or other organization for various purposes (Techopedia, 2020). Users typically browse the web by visiting the graph of links contained within web pages, typically starting at a human-maintained index of high-quality web pages such as Yahoo.com, or using a search engine (Brin and Page, 1998). Over time, Google has become the top search engine with the largest number of users worldwide, with a market share of 91% (GlobalStatCounter, 2022).

Research on search engines has been conducted. Chen et al. (2006) conducted a study entitled "Efficient Query Processing in Geographic Web Search Engines." In this study, they proposed a more efficient query processing algorithm than the query algorithms commonly used in geographic search engines. Furthermore, Allah et al. (2021) conducted a literature review on web search user interface (UI) design for the elderly, entitled "Designing a web search UI for the elderly community: a systematic literature review." This literature review provided suggestions for improving existing web search UIs to make them more senior-friendly, such as: clear and easily distinguishable visual displays, brief explanations of what will happen when a dialog button is pressed, search results pages that appear in a new window or tab, customizable character size and spacing in the search field, and more.Each search engine has a different

Universitas Negeri Jakarta
Phone: (021) 4890856
East Jakarta City
Jakarta, 13220

architecture. Google's architecture was chosen and used as a reference in the research topic of improving search engine architecture, which is the parent research of this research title, due to its superiority over other search engines. Furthermore, the research entitled "Search Engine Architecture Design by Integrating Web Crawlers, Page Ranking Algorithms, and Document Ranking" by Khatulistiwa (2022) combines the crawler module from Qoriiba's (2021) research, PageRank, and searchers from other previous studies into a console-based search engine (Khatulistiwa, 2022). The intuitive basis of PageRank is random walks on a graph. Consider an internet user as a "random surfer" who continuously clicks on subsequent links at random. However, if a user gets stuck in a webpage loop (a clicked link continues to display previously visited webpages), the user is unlikely to continue following the link; instead, they will immediately move to another page.

Research on distributed PageRank using the iterative aggregation-disaggregation (IAD) method with Block Jacobi smoothing has been conducted (Zhu et al., 2005). In simple terms, divide-and-conquer is performed by grouping web pages based on their domains, then calculating their local Pageranks and combining them using a memory-efficient communication method with a coordinator (Zhu et al., 2005). The result is a distributed Pagerank method that can be run on small main memory and converges faster, thus saving time (Zhu et al., 2005). Therefore, an alternative Pagerank algorithm that can be run on a single computer with limited main memory will be sought, by comparing the implementation of several Pagerank algorithms on a single computer.

METHOD

2.1 Research Design

There are several stages that must be completed to conduct this research. The research stages can be seen in Diagram 3.1. Several algorithms have been used that have not been explained previously, such as the Modified DPC (MDPC) and Random Walker.

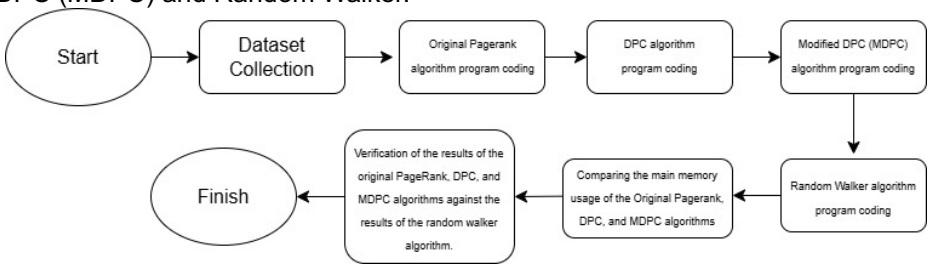

Figure 1: Research stage diagram

The MDPC algorithm was formulated due to the shortcomings of the DPC algorithm, which will be explained in the next section. The Random Walker algorithm, on the other hand, simulates the movement of web page visits and is the basis of the PageRank algorithm (Page et al., 1999), making it suitable as a reference for verifying results.

2.2 Data Collection

This study used data from the Khatulistiwa (2022) research database, supplemented by additional data collected using a crawling program. The obtained data was stored in a MySQL database, containing 13 tables or entities, the structure of which can be seen in the following tables:

Table 1: Crawling Table Structure

tfidf_word

No.	Atribut	Tipe Data
1.	id_word	int
2.	word	text
3.	page_id	int
4.	tfidf_score	double

tfidf

No.	Atribut	Tipe Data
1.	id_tfidf	int
2.	keyword	text
3.	page_id	int
4.	tfidf_total	double

pagerank

No.	Atribut	Tipe Data
1.	id_pagerank	int
2.	page_id	int
3.	pagerank_score	double

Universitas Negeri Jakarta
Phone: (021) 4890856
East Jakarta City
Jakarta, 13220

page_linking

No.	Atribut	Tipe Data
1.	id_linking	int
2.	page_id	int
3.	outgoing_link	text

page_tables

No.	Atribut	Tipe Data
1.	id_table	int
2.	page_id	int
3.	table_str	text

page_forms

No.	Atribut	Tipe Data
1.	id_form	int
2.	page_id	int
3.	form	text

page_images

No.	Atribut	Tipe Data
1.	id_image	int
2.	page_id	int
3.	image	text

crawling

No.	Atribut	Tipe Data
1.	id_crawling	int
2.	start_urls	text
3.	keyword	text
4.	total_page	int
5.	duration_crawl	time
6.	created_at	timestamp

page_scripts

No.	Atribut	Tipe Data
1.	id_script	int
2.	page_id	int
3.	script	text

page_list

No.	Atribut	Tipe Data
1.	id_list	int
2.	page_id	int
3.	list	text

page_styles

No.	Atribut	Tipe Data
1.	id_style	int
2.	page_id	int
3.	style	text

page_information

No.	Atribut	Tipe Data
1.	id_page	int
2.	crawl_id	int
3.	url	text
4.	html5	tinyint
5.	title	text
6.	description	text
7.	keywords	text
8.	content_text	text
9.	hot_url	tinyint
10.	size_bytes	bigint
11.	model_crawl	text
12.	duration_crawl	time
13.	created_at	timestamp

Of the 13 tables, the ones that will be used are the page_linking, page_information, and pagerank tables. The page_linking table contains information about which page the link originates from via the page_id attribute and where the link points to via the url attribute. For the page_information table, the only attributes used are page_id and url. Once the calculation is complete, the results are stored in the pagerank table.
This study used two datasets. The first dataset, hereinafter referred to as Dataset 1, is a combined dataset of the dataset obtained from the Khatulistiwa (2022) study and additional data obtained through crawling. Previously, Dataset 1 had no more than 11,000 rows of web pages, with page_information now containing 20,493 rows and page_linking containing 2,915,842 rows. The page_information data in Dataset 1 can be grouped into 560 clusters based on their domains, as seen in Table 3.13.

Table 2: Cluster data in the Dataset

No.	Domain	Jumlah Halaman
1	detik.com	2.215
2	unj.ac.id	2.208
3	sport.detik.com	1.279
4	finance.detik.com	1.098
5	repository.unj.ac.id	1.089
6	news.detik.com	802
7	oto.detik.com	779
8	inet.detik.com	671
9	support.google.com	630
10	food.detik.com	626
⋮	⋮	⋮
558	codingcompetitions.withgoogle.com	1
559	googledevelopers.blogspot.com	1
560	skillshop.exceedlms.com	1

As seen in Table 3.13, web pages are dominated by the domains "detik.com" and "unj.ac.id" and their subdomains. This is because during crawling, the initial web pages crawled were "detik.com" and

Universitas Negeri Jakarta
Phone: (021) 4890856
East Jakarta City
Jakarta, 13220

"unj.ac.id." This also demonstrates the tendency for web pages to have intra-links, that is, links that point to other pages within the same domain, which is one of the bases of the DPC algorithm (Zhu et al., 2005). Second, Dataset 2 is a small dataset and small domains that were intentionally collected to observe the difference in algorithm performance between datasets containing many large domains and datasets containing small domains. The limit for each domain used when collecting data was 20 web pages per domain. Dataset 2 contains 100 rows of page_information, 5,944 rows of page_linking, and 5 clusters or domains. The cluster data in Dataset 2 can be seen in Table 3.14.

Table 3: Cluster data in Dataset 2

No.	Domain	Jumlah Halaman
1	unj.ac.id	20
2	ppid.unj.ac.id	20
3	fip.unj.ac.id	20
4	fbs.unj.ac.id	20
5	fmipa.unj.ac.id	20

2.3 Weaknesses of the Original PageRank Algorithm

The Original PageRank algorithm works by iteratively multiplying the web page ranking vector by the transition matrix of the website graph. The problem arises because this transition matrix requires a significant amount of main memory, with a complexity of O(N2). For example, if the data type used is 64-bit floating point and the dataset contains 10,000 web pages, the resulting transition matrix is a square matrix measuring 10,000 × 10,000, and the main memory required is 10,000 × 10,000 × 64 bits = 6,400,000,000 bits = 800 Megabytes. While the previous example is quite small, in reality, the internet has billions of web pages. If the Original PageRank algorithm is used without special handling, it would consume petabytes or even exabytes of memory. Consequently, if implemented on a personal computer with only 4 to 32 Gigabytes of memory, the program would crash.

2.4 Further Explanation of the DPC Algorithm

The algorithm begins by inputting a transition matrix P and a set of web page clusters G. Both are obtained from the Khatulistiwa (2022) research database, composed of two entities that will be explained later in the next section. The damping factor d follows the research of Zhu et al. (2005), i.e., d = 0.85, and a value of $0 < \epsilon < 1$ for error tolerance.Next, for each cluster, a local transition matrix is created. Then, calculate the local PageRank value by inputting Qi and the error tolerance value ϵ. The resulting initial local PageRank is a vector with dimensions N × 1.

The next step is a loop. k is the number of iterations. An aggregate matrix is created. R is an n × N matrix, where n is the length of Gi and N is the total number of web pages. P is the overall transition matrix,with dimensions N × n, and the disaggregation matrix N × n. The matrix Ak is used as the input transition matrix for the coarse-grained PageRank calculation. It should be noted that since the dimension of Ak is n × n, the dimension of the vector is n × 1. Consequently, the dimension of the vector zk is n × 1. This step is called the coarse-level roughing solution (Zhu et al., 2005). Next, for each cluster Gi, create an augmented (Ni + 1) × (Ni + 1) local transition matrix Bk i. In the upper left part of the matrix Bk i , Pi i is the local transition matrix of cluster Gi. In the lower left part, there is a multiplication of the row vector eT with dimension 1 × N and P∗i, the transition matrix from the member page of Gi to the member page of G with dimension N × Ni. The final result of this multiplication is a row vector with dimension 1 × Ni that will occupy the bottom row of the matrix Bk i along with the scalar αk, which becomes the bottom and rightmost element of the matrix Bk i . The value of αk depends on the total number of rightmost columns of matrix Bk i, which is equal to 1. In the upper right corner, there is the matrix Pi∗, with dimensions Ni × N and also the transition matrix from page G to page Gi, multiplied by the disaggregation matrix S(πk) and the vector zk. The product of these three matrices and vectors is then subtracted from the product of matrix Pii, vector πk i, and scalar z i. The result of this subtraction is divided by the subtraction of 1 minus zk i. The final result of the operation on the

Universitas Negeri Jakarta
Phone: (021) 4890856
East Jakarta City
Jakarta, 13220

upper right corner of matrix Bk i is a column vector of Ni × 1, along with αk, the rightmost column of matrix Bk i. The transition matrix, vector i e.
Next, calculate the Pagerank algorithm with input matrix Bk i as (Ni + 1) as the initial ranking vector, and the value of ϵ. The final PageRank result is a column vector of N × 1. The first row through N is the vector ωk+1, and the (N+1)th row is the scalar value βk+1i. This step is referred to as smoothing at a finer level (Zhu et al., 2005). Afterward, the page ranking value in each cluster, denoted by π~k+1i, is updated using the calculation in Equation 8. The page ranking is then normalized using Equation 9. Calculate the difference between the current iteration's web page ranking and the previous iteration's. If it is less than the error tolerance ϵ, the algorithm returns the value of πk+1. Otherwise, repeat the loop.

2.5 Weaknesses of the DPC Algorithm

In the DPC algorithm, obtaining matrix A requires multiplying the R matrix, the P matrix, and the R(π) matrix. Multiplying the P matrix as a whole creates problems and consumes a significant amount of main memory, a problem similar to that encountered in PageRank Original. To overcome this, special handling is required during the multiplication to fit the main memory. Consequently, the algorithm runs slower due to the process of splitting the P matrix into smaller ones. In this study, we split the P matrix into its column vectors when multiplying the R matrix by the P matrix.
The essence of steps four through seven of the DPC algorithm is to find the domain ranking and adjust the domain ranking to the ranking of web pages that are still isolated in their respective domains. The A matrix is essentially the domain transition matrix, while the z vector is the domain ranking vector. These steps can be simplified; in this study, a modified DPC algorithm, called the Modified DPC (MDPC), is formulated.

2.6 Modified DPC Algorithm

Previously, we discussed two algorithms: Original PageRank in the research of Page et al. (1999) and Distributed PageRank Computation in the research of Zhu et al. (2005). A modified DPC algorithm, or Modified DPC (MDPC), was proposed. In general, MDPC simply reduces the steps in the DPC algorithm, based on the main idea of DPC: separately calculating web page rankings based on their domains and combining them by calculating the domain rankings themselves. The main problem with the DPC algorithm is the step in obtaining matrix A in step 2.10, which involves multiplying the entire P matrix.

2.6 Random Walker Algorithm

The intuitive basis of the PageRank algorithm is a random walk on a graph. A simplified version is a random walk probability distribution on a web graph (Page et al., 1999). A program will be created to simulate this random walk process, called the Random Walker Algorithm. The Random Walker Algorithm is used to compare the calculation results of the DPC, MDPC, and original PageRank algorithms. The steps in the Random Walker Algorithm can be seen in Algorithm 6.In the first step, the number of iterations is necessary because, fundamentally,unlike the PageRank algorithm, which can converge, the Random Walker algorithm cannot terminate except by limiting the number of iterations.The second stepsimulates that internet users can come from any web page. The third step determines the probability of a walker moving from the initial web page to another page. This P matrix also contains the probability of a walker jumping to another web page that is not linked to each other or is not directly connected.

2.6 Development Stages

An assessment of the main memory efficiency of the PageRank, DPC, and MDPC algorithms will be conducted. This research is a sub-research of the main research on search engine development. Previous research by Qoriiba (2021) developed a crawler module and other supporting modules. Furthermore, Khatulistiwa (2022) combined the Qoriiba (2021) crawler module, Google PageRank, and a TF IDF-based search engine into a console-based search engine. Concurrently, Pratama (2022) created an indexer module using an Induced Generalized Suffix Tree, and Zalghornain (2022) used Continuous-Bag-of-Words and Continuous Skip-Gram models.
Because this research is a sub-research, the technology stack used in its implementation will be the same as the main research, using the Python 3 programming language and a MySQL database.The algorithm implementation will follow the algorithms described previously.

Universitas Negeri Jakarta
Phone: (021) 4890856
East Jakarta City
Jakarta, 13220

The estimated completion time is one month, and it is possible that it will take longer or shorter depending on the difficulties and complexities encountered. To test the memory usage, execution time, and results of the web page ranking algorithms, the following steps were taken:

1. Code four algorithm programs: Pagerank Original,
2. DPC, MDPC, and Random Walker.
3. Determine the input values for the damping factor, error tolerance, and the same dataset.
4. Run the Pagerank Original, DPC, MDPC, and Random Walker programs on dataset 1 (20,493 pages).
5. Pagerank Original, DPC, and MDPC.
6. Compare the execution time and main memory usage of the algorithms.
7. Verify the similarity of the results between the Pagerank Original, DPC, and MDPC algorithms and the Random Walker algorithm.
8. Run the Pagerank Original, DPC, MDPC, and Random Walker programs on dataset 2 (100 pages).
9. Compare the execution time and main memory usage of the Pagerank Original, DPC, and MDPC algorithms.
10. Verify the similarity of the results between the Pagerank Original, DPC, and MDPC algorithms and the Random Walker algorithm.

2.6 Comparison Method

This study compared three things: the first was the main memory usage, the length of time the algorithms ran, and the similarity of the results produced by the three algorithms. There was no specific method for comparing main memory usage and execution time. The three algorithms were simply run and then the main memory usage and execution time of each algorithm could be observed and compared. To assess the similarity of the results, this study followed the method used by Zhu et al. (2005).

RESULTS AND DISCUSSION

3.1 Result

After running the program on all datasets, the following results were obtained. In Dataset 2, the peak memory usage for all three algorithms was relatively similar, due to the relatively small data compared to the program's memory overhead. Furthermore, in Dataset 1, the largest peak memory usage occurred in the original PageRank algorithm, at 3.4 GB, starting when the 20,493 × 20,493 matrix P was formed. Since each matrix cell is a 64-bit floating-point decimal number, the size of the P matrix in memory is approximately 3.4 GB.

Furthermore, the DPC algorithm had the largest memory usage, at 842 MB, which occurred during the formation and caching of the P∗i matrix. Considering the largest domain in Dataset 1 is "detik.com," with 2,215 pages, the largest P∗i matrix dimension is 20,493 × 2,215, which will consume approximately 363 MB of memory. When cached using the pickle library, objects are copied before being written to the cache, resulting in memory usage of 2 × 363 MB = 726 MB.

In the MDPC algorithm, the peak memory usage is approximately 86 MB, with peak memory usage occurring during the formation and cache storage of the Q matrix. Since the Q matrix is a local transition matrix for a domain, the largest Q matrix is the Q matrix for the "detik.com" domain, with 2,215 pages. The resulting Q matrix size is 2,215 × 2,215, which in terms of memory size is approximately 39 MB. During the cache storage process, a temporary object copy is performed, doubling the memory size to approximately 86 MB. A comparison of the peak memory usage and execution time of the Original Pagerank algorithm, the Distributed Pagerank Computation (DPC) algorithm, and the Modified DPC (MDPC) algorithm can be seen in Table 4.

Table 4: Peak memory usage and execution time

Algoritma	Puncak Penggunaan Memori(*Mega Byte*)	Waktu(Detik)
Dataset 1 (20.493 halaman)		
Pagerank	3.417,63239	328,803
DPC	842,48153	1.151,628
MDPC	86,58194	816,375

Universitas Negeri Jakarta
Phone: (021) 4890856
East Jakarta City
Jakarta, 13220

Dataset 2 (100 halaman)		
Pagerank	2,04511	0,568
DPC	2,0563	19,372
MDPC	2,05595	0,652

Next, the similarity value between the PageRank values generated by the original PageRank program, DPC, MDPC, and RandomWalker will be calculated. PageRank values are vectors and sorted by page ID. In Dataset 1, the KDist value for each web page ranking vector generated by the PageRank, DPC, MDPC, and RandomWalker algorithms can be seen in Table 4.2.

Table 5: Kendall distance values for PageRank vectors in Dataset 1 (20,493 pages)

	Pagerank	DPC	MDPC	Random Walker
Pagerank	0,0	0,24956	0,25985	0,02716
DPC	0,24956	0,0	0,27681	0,25208
MDPC	0,25985	0,27681	0,0	0,26387
Random Walker	0,02716	0,25208	0,26387	0,0

The most similar vectors, or those with the smallest KDist values, are the Pagerank and RandomWalker vectors, with a difference of 0.02716, or a 2.7 percent difference in order. Meanwhile, the Kdist values for DPC against Pagerank and RandomWalker, respectively, are 0.24956 and 0.25208, or a 25 percent and 25.2 percent difference in order. Furthermore, the Kdist values for MDPC against DPC, Pagerank, and Random Walker, respectively, are 0.27681, 0.25985, and 0.26387, or a 27.7 percent, 26 percent, and 26.4 percent difference in order.

In Dataset 2, the Kdist values for each web page ranking vector generated by the Pagerank, DPC, MDPC, and Random Walker algorithms can be seen in Table 4.3. The most similar vectors, or those with the smallest Kdist values, are the Pagerank and Random Walker vectors, with a difference of 0.03293, or a 3.3 percent difference in order. Meanwhile, the Kdist values for DPC versus Pagerank and Random Walker are 0.17131 and 0.19131, or a difference of 17.1 percent and 19.1 percent, respectively. Furthermore, the Kdist values for MDPC versus DPC, Pagerank, and Random Walker are 0.11091, 0.12586, and 0.14949, or a difference of 11.1 percent, 12.6 percent, and 14.9 percent, respectively.

Table 6: Kendall's distance values for Pagerank vectors in dataset 2 (100 pages)

	Pagerank	DPC	MDPC	Random Walker
Pagerank	0,0	0,17131	0,12586	0,03293
DPC	0,17131	0,0	0,11091	0,19131
MDPC	0,12586	0,11091	0,0	0,14949
Random Walker	0,03293	0,19131	0,14949	0,0

In addition to differences in ranking values, differences in the value or reduction results between each web page between algorithms can be seen in Tables 4.8 and 4.9 for Dataset 1, and Tables 4.10 and 4.11 for Dataset 2. Understanding the differences in values is important to determine the extent of the deviation or change in ranking results between the four algorithms (PagerankOriginal, DPC, MDPC, and RandomWalker). The larger the difference, the greater the difference. This is also inevitable due to the differences in the algorithms used. However, generally, differences below 10-5 are tolerable and considered the same. Therefore, the values shown in the table are only five decimal places, or five decimal places.

Universitas Negeri Jakarta
Phone: (021) 4890856
East Jakarta City
Jakarta, 13220

Table 7: Difference in webpage ranking values Dataset

Dataset 1 (20.493 halaman)				Dataset 2 (100 halaman)			
page_id	Pagerank - DPC	Pagerank - MDPC	DPC - MDPC	*page_id*	Pagerank - DPC	Pagerank - MDPC	DPC - MDPC
1	0,00002	0,00001	0,0	1	0,00107	0,00128	0,00021
2	0,00245	0,00008	0,00237	2	0,00088	0,00112	0,00024
3	0,00245	0,00008	0,00237	3	0,00156	0,00314	0,0047
4	0,00251	0,00217	0,00034	4	0,00571	0,00079	0,00492
5	0,00244	0,0019	0,00054	5	0,00571	0,00079	0,00492
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
20.490	0,00006	0,00001	0,00005	96	0,00269	0,00257	0,00012
20.491	0,00003	0,00003	0,0	97	0,00784	0,00038	0,00746
20.492	0,00003	0,00003	0,0	98	0,00328	0,00482	0,0081
20.493	0,00006	0,00001	0,00005	99	0,00328	0,00348	0,0002
20.494	0,00003	0,00003	0,0	100	0,00277	0,00244	0,00033

3.7 Discussion

The significant difference in results between Distributed Pagerank Computation (DPC) and Modified DPC (MDPC) versus Pagerank Original and Random Walker is due to the different calculation principles. DPC and MDPC separate web pages based on their domains, while Pagerank and Random Walker perform the calculation as a whole.

In conclusion, the Pagerank Original algorithm is the fastest and most similar in results to the Random Walker algorithm in ranking web pages, but requires more main memory. Meanwhile, the DPC and MDPC algorithms require less main memory, making them suitable for single-machine computers with limited main memory. However, this results in significantly different results compared to the Pagerank Original algorithm and the Random Walker algorithm, as well as slower execution times.

CONCLUSION

Based on the implementation and testing results of the Pagerank, DPC, MDPC, and Random Walker algorithms, the following conclusions were obtained:

1. The Pagerank algorithm is an algorithm for calculating web page rankings based on a random walk on a web page graph (Page et al., 1999).The Pagerank algorithm has the problem of large memory usage.The DPC algorithm, which uses the divide-and-conquer method (Zhu et al., 2005), is used to address the Pagerank algorithm's problems. The MDPC algorithm is a modification of the DPC algorithm formulated in this study because some steps in the DPC algorithm can be simplified. A Random Walker simulation program was created to compare the results of the Pagerank, DPC, and MDPC algorithms.
2. The test results show that the fastest web page ranking algorithm in terms of execution time is the Pagerank algorithm, while in terms of peak memory usage, MDPC and DPC are much smaller than the Pagerank algorithm. However, after testing the ranking results by calculating the KDist values between each algorithm, the results from the PageRank algorithm were very similar to those from the Random Walker algorithm, compared to the DPC algorithm, and MDPC to Random Walker.Therefore, it can be concluded that the DPC and MDPC algorithms are suitable for single-machine computers with limited main memory, but at the expense of similar results and slower execution times.

REFERENCES

1. Allah, K. K., Ismail, N. A., and Almgerbi, M. (2021). Designing web search ui for the elderly community: a systematic literature review. Journal of Ambient Intelligence and Humanized Computing.
2. Armstrong, M. (2021). How many websites are there? https://www.statista.com/chart/19058/number-of-websites-online/. on 07-30-2022.

Universitas Negeri Jakarta
Phone: (021) 4890856
East Jakarta City
Jakarta, 13220

3. Berners-Lee, T., Cailliau, R., Groff, J.-F., and Pollermann, B. (1992). World-wide web: The information universe. Internet Research, 2(1):52–58.
4. Brin, S. and Page, L. (1998). The anatomy of a large-scale hypertextual web search engine.
5. Chen, Y.-Y., Suel, T., and Markowetz, A. (2006). Efficient query processing in geographic web search engines. In SIGMOD '06: Proceedings of the 2006 ACM SIGMOD international conference on Management of data, pages 277–288.
6. Courtois, P. and Semal, P. (1986). Block iterative algorithms for stochastic matrices. Linear Algebra and its Applications, 76:59–70.
7. GeeksForGeeks (2022). Introduction https://www.geeksforgeeks.org/introduction-of-b-tree-2/. of b-tree. Accessed 09-28-2022.
8. GlobalStatCounter (2022). 2022. Search engine market share worldwide - July https://gs.statcounter.com/search-engine-market-share#monthly-200901 202208-bar. Accessed September 28, 2022. 85 86
9. Huss, N. (2022). How many websites are there in the world? (2022). https://siteefy.com/how-many-websites-are-there/. Accessed September 28, 2022.
10. Khatulistiwa, L. (2022). Designing a search engine architecture by integrating web crawlers, page ranking algorithms, and document ranking.
11. Kontovasilis, K. P. and Mitrou, N. M. (1995). Markov-modulated traffic with nearly complete decomposability characteristics and associated fluid queuing models. Advances in Applied Probability, 27(4):1144–1185.
12. Mishra, D. (2016). Proper weak regular splitting and its application to convergence of alternating iterations.
13. MySQLDoc (2022). How https://dev.mysql.com/doc/refman/8.0/en/mysql-indexes.html. 07-30-2022. 31(3):265–279. mysql uses Accessible indexes.
14. Neumann, M. and Plemmons, R. J. (1978). Convergent nonnegative matrices and iterative methods for consistent linear systems.
15. Numerische Mathematic, Page, L., Brin, S., Motwani, R., and Winograd, T. (1999). The pagerank citation ranking: Bringing orders to the web. Technical Report 1999-66, Stanford InfoLab.
16. Pratama, Z. (2022). Designing an indexing module in a search engine using an induced generalized suffix tree for document ranking.
17. Qoriiba, M. F. (2021). Designing a crawler as a support for search engines. Sample, I. (2018). What is the internet? 13 key questions answered. https://www.theguardian.com/technology/2018/oct/22/what-is-the-internet-13 key-questions-answered. Accessed July 30, 2022. 87
18. Seymour, T., Frantsvog, D., and Kumar, S. (2011). History of search engines. International Journal of Management & Information Systems (IJMIS), 15(4):47. Techopedia (2020). Website. https://www.techopedia.com/definition/5411/website. Accessed July 30, 2022.
19. Wilson, R. J. (1996). Introduction to Graph Theory. Longman Group Ltd. Zalghornain, M. (2022). Design of a text retrieval system using the continuous-bag-of-words model and the continuous skip-gram model on a document collection.
20. Zhu, Y., Ye, S., and Li, X. (2005). Distributed PageRank computation based on iterative aggregation-disaggregation methods. In CIKM '05: Proceedings of the 14th ACM International Conference on Information and Knowledge Management, pages 578–585.