Top-K Query for Large Dataset of Restaurant Review Based-on Hadoop MapReduce Framework

Ni Nyoman Putri Utami[1], Heri Wijayanto[*, 1], I Gde Putu Wirarama[1]
[1]Dept Informatics Engineering, Mataram University
Jl. Majapahit 62, Mataram, Lombok NTB, INDONESIA
Corresponding Author: heri@unram.ac.id

To develop culinary tourism post-COVID-19, high-quality facilities and services are necessary. Information technology can assist through a top-k query-based decision-making system. This study implements top-k queries on a distributed Hadoop MapReduce system to evaluate its ability to manage data and select culinary tourism potential. Results indicate that multi-node execution with 3 nodes and a large number of dimensions is faster for processing large data sets over 1 million, while single-node execution is faster for relatively small data sets. Using 6 nodes for processing 20 million data with 5 dimensions is the optimal method with the shortest execution time. By utilizing information technology and a top-k query-based decision-making system, the development of culinary tourism potential can be carried out more efficiently and effectively. The performance of MapReduce in processing culinary tourism potential data can be optimized by using multi-node execution for large data sets and single-node execution for relatively small data sets.

***Key words*: Hadoop MapReduce, Top-k Query, Distributed System, Tourism Potential, Big Data.**

I. Introduction

Tourism is one of the sectors that serves as the backbone of the economy in many countries around the world. In 2022, despite the fact that life has entered a new normal of the COVID-19 pandemic transition phase, the global tourism industry has bounced back to recover the world economy. According to a survey conducted by UNWTO, the tourism sector has recovered nearly 60% of pre-pandemic levels as various countries have lifted COVID-19-related restrictions since September 19th, 2022. It is estimated that approximately 474 million international tourists traveled during the pre-pandemic period in the first 7 months of 2022 [1].

Culinary tourism is an integral part of the tourism industry, as food, particularly local cuisine, has a distinctive appeal in terms of appearance, taste, and aroma that is well-known and enjoyed. This ultimately encourages tourists to revisit and recommend a destination to others. Therefore, high-quality attractions, facilities, and services for culinary tourism need to be considered in order to enhance tourist appeal, and thereby, allow tourism potential to continue to grow.

In the era of information technology, the decision-making system for recommending potential tourist destinations can be based on top-k query. In the age of information technology, a top-k query-based decision-making system can be used to suggest potential tourism sites. The top-k query method uses a specific scoring formula to determine the top-k objects and a set of k data that best match the user's preferences to inform decision-making [2]. The object score serves as an assessment of the object based on its traits. To calculate the overall object score based on the processing and management of the data, top-k query calculations can be used on a distributed system when dealing with a big amount of data.

Distributed data processing can be performed using Hadoop MapReduce. Hadoop is one of the technologies that enables the collection and distributed processing of large amounts of data on a cluster of interconnected computers with a simple model [3]. Meanwhile, MapReduce is a parallel programming paradigm that exists within Hadoop and is useful for processing large amounts of data between each node [4].

Therefore, this research aims to investigate the performance of MapReduce in implementing top-k query calculations for managing data on the selection of potential culinary tourism destinations based on restaurant reviews. It is expected that this research will increase the potential of culinary tourism in a region and provide a prototype for analyzing which culinary tourism potentials need to be developed for local government.

II. Literature Review

A. Related Research

In a research conducted by Ahmad Luky Ramdani in 2016 entitled "Twitter Account Selection Using Skyline Query in MapReduce Framework," the researcher identified influential Twitter accounts based on OL characteristics using MT features and sentiment analysis in the MapReduce framework. The skyline query algorithm was then applied in the MapReduce framework to select influential accounts. Based on these characteristics, 16 influential accounts were obtained from a total of 65,702 accounts in the data set. The identification and selection process for accounts in the MapReduce framework consistently showed faster execution times compared to conventional models [5].

In the research conducted by A. Muh. Ryanto in 2017 entitled "Performance Analysis of Big Data Framework on Virtualized Cluster: Hadoop MapReduce and Apache Spark", the researcher implemented virtualization technology on the Hadoop MapReduce and Apache Spark clusters to determine the performance of both technologies on a virtualized cluster. The computational performance results showed that Apache Spark was 3 to 5 times faster on a single-node virtualized cluster and 1 to 4 times faster on a multi-node virtualized cluster than the performance of Hadoop MapReduce. In the I/O cluster performance test, the throughput generated was higher when Apache Spark was used together [6].

In the research conducted by Ecko Fernando Manalu in 2009 entitled "Analysis of Skyline Query and Top-K Query in Context Preference Aware Service", the researcher applied skyline query and top-k query to Context Preference Aware (CPA), a type of geographic information system that provides relevant information services to users based on their context and preferences. The result of this research is the layered processing of query results using hierarchical filtering with top-k and skyline queries as algorithmic strategies for solving problems to provide relevant information. This CPA service is very advanced and sophisticated for the present time, although it has not been implemented in Indonesia due to developer and technology constraints [7].

In the research conducted by Dony Rusdiyatno et al. titled "Alternative Route Information System Using Top-k Query-based WEB on BlackBerry™", the researchers applied top-k query to inform alternative routes on BlackBerry™ 9300 3G smartphones by conducting 15 experiments with 44 nodes divided into 94 segments. The testing was evaluated based on the accuracy of the generated routes and the speed of displaying alternative routes, resulting in a search accuracy rate of 93% with an average processing time of 7.3 seconds [8].

In the research conducted by Wijayanto et al. in 2022, entitled "LShape Partitioning: Parallel Skyline Query Processing using MapReduce", the authors applied a two-phase MapReduce Skyline processing that utilizes a new LShape partitioning strategy, built in the first phase with lightweight MapReduce computation. In the second phase, LShape partitions are broadcasted to mappers in parallel to compute local skyline and their merge is global skyline with lightweight merging process. This strategy is a MapReduce skyline processing with a multiple reducer approach. The LShape partitioning strategy has the advantage of utilizing a new filtering method called propagation pruning. The LShape partitioning and propagation pruning strategy perform better than the state-of-the-art non-sampling algorithm approaches MR-GPMRS and PPFPGPS. This was demonstrated in intensive experiments on anti-correlated, independent, and correlated datasets that performance improves in high-dimensional and high data volume settings [9].

In a research conducted by Wijayanto et al. in 2021 titled "Upgrading Product Based on Existing Dominant Competitor", the researchers applied the Top-Down Recursive Depth First Search Algorithm (TDRDFS) method to construct the points of Dominant Graph of Intersection (DGI) for the dominance area model, so that in this study, the expected number of customers can be determined using DGI for product improvement recommendations. This applies to improving products in Industry 4.0 where customer preferences change when new products are introduced to the market. This phenomenon motivates manufacturers to compete in creating innovations to take over the market [10].

Therefore, the difference of this research from previous related studies is that this study will analyze the performance of the distributed system Hadoop MapReduce for recommending potential tourism destinations that need to be developed by implementing top-k query calculations on Hadoop MapReduce. The system's performance will be tested using data from open data and synthetic data with various file sizes, as well as the number of nodes used in the Hadoop Cluster.

B. Supporting Theory

B.1. Hadoop cluster

Hadoop is one of the technologies that enables the collection and distributed processing of large data sets across a set of interconnected computers (cluster computers) with a simple model [3]. The architecture of a Hadoop Cluster is illustrated in Fig. 1.

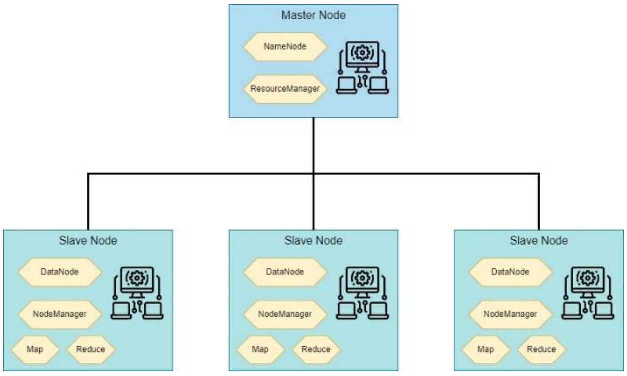


Fig. 1. Hadoop cluster architecture

Based on Fig. 1, the architecture of a Hadoop Cluster consists of one Master Node and a number of Slave Nodes. The Master Node has the components Name Node and Resource Manager, while the Slave Nodes have Data Node, Node Manager, and MapReduce. The Name Node and Data Node in the Hadoop architecture are HDFS (Hadoop Distributed File System) that serve as data storage. Then, the Resource Manager and Node Manager in the Hadoop architecture are YARN (Yet Another Resource Manager) that coordinate nodes precisely so that applications and a large number of users can share resources effectively [3].

B.2. MapReduce

MapReduce is a parallel programming paradigm in Hadoop that is useful for processing large amounts of data between nodes. Each job that is executed is mapped from the name node to the data node and executed by MapReduce. The MapReduce operations available in Hadoop include wordcount. Wordcount is a program for counting words in plaintext files [4].

The wordcount process divides the input data into several evenly sized parts according to the number of mappers used. Then, the map function processes each word and produces key-value pairs. All key-value pairs from the map function are combined and grouped based on their key-value pairs for sorting. After that, they are passed to the reduce function, which produces the final output [6].

B.3. Top-k query

Top-k query is a common method for decision-making based on a set of k data that best match user preferences and a specific scoring function [2]. To identify top-k objects, all objects are evaluated based on several scoring functions. An object's score acts as an evaluation for the object based on its characteristics (e.g., price, distance, and size of the object in the database or color and texture of images in a multimedia database). Data objects are usually evaluated with multiple score predicates that contribute to the total object score [11]. The highest-scored evaluations form the output of the top-k query [12]. Several applications benefit from top-k queries, including web search, digital libraries, and e-commerce [2]. For example, to use top-k query calculations, one can use the sample data in Table 1 to determine which potential tourism areas need to be developed.

TABLE I. EXAMPLE OF TOURIST AREAS

Daerah Wisata	Penilaian Makanan (a_1)	Penilaian Suasana (a_2)	Jumlah Total Ulasan (a_3)
Italy	4	5	36
France	5	3	25
Spain	5	4	126

Based on the data in Table 1 that has 3 attributes (characteristics), they can be assumed to be a_1, a_2, and a_3 for each attribute. To calculate the total score of the object, Eq. 1 can be used as follows.

$$s = w_1 \times a_1 + w_2 \times a_2 + \cdots + w_i \times a_i \qquad (1)$$

Description:

s : total score of the object

w_i : weight for each attribute i

a_i : value of attribute i

Assuming that the weight of each attribute is 1, the calculation result with Eq. 1 for these three regions can be seen in Table 2. Therefore, the region that needs to be developed is France, as it has a smaller total object score compared to Italy and Spain.

TABLE II. EXAMPLE RESULT OF TOP-K QUERY CALCULATION

Daerah Wisata	$w_1 \times a_1$	$w_2 \times a_2$	$w_3 \times a_3$	s
Italy	4	5	36	45
France	5	3	25	33
Spain	5	4	126	135

B.4. Normalization

Normalization is the process of scaling attribute values of data to fit into a predetermined range. The following are some normalization techniques [13]:

- Min-Max Normalization: Min-Max normalization is a normalization method that applies a linear transformation to the original data to balance the value ratio before and after processing. This method can use the following equation:

$$normalized(x) = \frac{minRange+(x-minValue)(maxRange-minRange)}{maxValue-minValue} \qquad (2)$$

- Z-score Normalization: Z-score normalization is a normalization method based on the mean and standard deviation of the data. This method is very useful when the actual minimum and maximum values of the data are unknown. The equation used is as follows:

$$newvalue = \frac{oldvalue-mean}{stdev} \qquad (3)$$

- Decimal Scaling Normalization: Decimal scaling is a normalization method that moves the decimal value of the data towards a desired direction. The formula used is as follows:

$$newvalue = \frac{oldvalue-mean}{10^i} \qquad (4)$$

III. RESEARCH METHODOLOGY

A. Tools and Materials

A.1. Tools

The following are the software and hardware tools used in this research:

- An Asus X441N Laptop with Intel Inside CPU Intel 2Core N3350 up to 2.4GHz RAM 4GB.
- 8 HP PCs used for Hadoop cluster purposes with Intel Core i5 CPU 3.00GHz RAM 4GB.
- Windows 10 Operating System.
- Java Programming Language.
- Hadoop MapReduce framework.
- Microsoft Excel 2016.
- Microsoft Word 2016.
- LibreOffice Calc.

A.2. Materials

The materials used for data processing in making decisions to determine potential tourism areas that need to be developed are data obtained from open data sources and synthetic data. Data obtained from open data sources is obtained from Kaggle, which is the European restaurants data according to TripAdvisor. TripAdvisor is the most popular travel website and stores data for almost all restaurants, showing the location (even latitude and longitude coordinates), restaurant description, user ratings and reviews, and many other aspects [14]. However, in this research, only the following attributes from the European restaurants data were used: average restaurant rating, total reviews count, food rating, service rating, and atmosphere rating. Meanwhile, synthetic data is data obtained by generating data according to the desired data size.

B. Research Flow

The research flow on the implementation of top-k query calculation in distributed Hadoop MapReduce system for culinary tourism potential recommendation based on restaurant reviews consists of several stages, namely literature review, system design, system implementation, system testing, and documentation.

- Literature review, this stage is the initial stage of this research which is carried out by searching for reference journals related to the research topic, namely Hadoop, MapReduce, wordcount operation, and top-k query.
- Data collection, at this stage data collection is carried out which will be used for data processing and analysis in the research.
- System design, at this stage, the concept of system design is made, such as the workflow of the system.
- Data normalization, at this stage, the process of scaling attribute values of obtained data is performed to fit within the determined range.
- System testing, at this stage, system testing is carried out, where if the system is able to run according to the requirements, it will proceed to the next stage. However, if the system is unable to run as needed, the system will be fixed based on the system design.
- Documentation, at this stage, documentation is carried out in the form of a report on the results of the research conducted.

C. System Design

The system design in this research uses a model as shown in Figure 4. The obtained data will be processed to generate key-value pairs for each data. Then, the value of the data attribute will be identified using top-k query calculation on the MapReduce framework to determine the final value or score of each key's object.

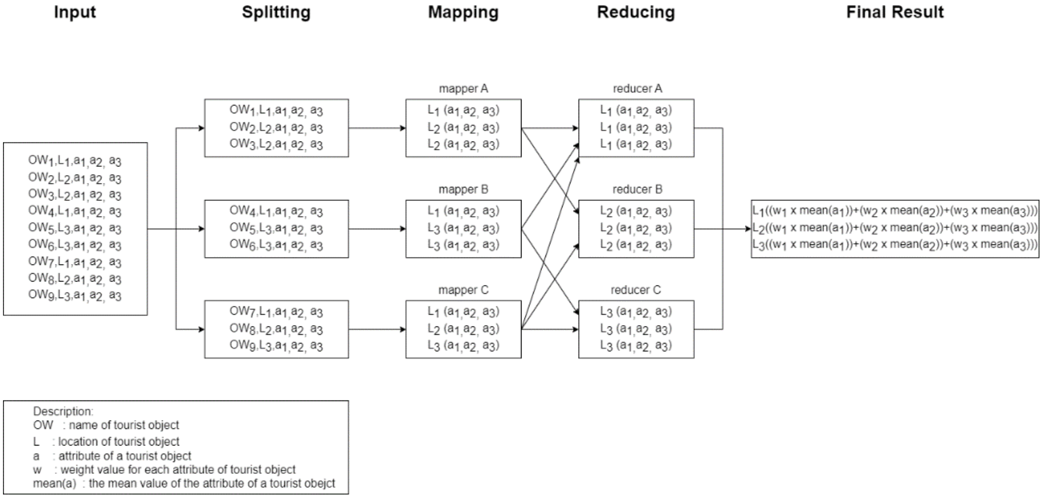

Input
Splitting
Mapping
Reducing
Final Result
mapper A
mapper B
mapper C
reducer A
reducer B
reducer C
OW1,L1,a1,a2, a3
OW2,L2,a1,a2, a3
OW3,L2,a1,a2, a3
OW4,L1,a1,a2, a3
OW5,L3,a1,a2, a3
OW6,L3,a1,a2, a3
OW7,L1,a1,a2, a3
OW8,L2,a1,a2, a3
OW9,L3,a1,a2, a3
L1 (a1,a2, a3)
L2 (a1,a2, a3)
L3 (a1,a2, a3)
L1((w1 x mean(a1))+(w2 x mean(a2))+(w3 x mean(a3)))
L2((w1 x mean(a1))+(w2 x mean(a2))+(w3 x mean(a3)))
L3((w1 x mean(a1))+(w2 x mean(a2))+(w3 x mean(a3)))
Description:
OW : name of tourist object
L : location of tourist object
a : attribute of a tourist object
w : weight value for each attribute of tourist object
mean(a) : the mean value of the attribute of a tourist obejct

Fig. 2. System design

- Input, at this stage, the input to the system consists of the obtained data of tourism objects and their attributes.
- Splitting, the tourism object data and its attributes are divided evenly according to the number of mappers that have been determined.
- Mapping, at this stage, the tourism objects that have been split in the previous stage are mapped to determine the number of tourism objects in each mapper, resulting in key-value pairs.
- Reducing, at this stage, tourism objects are grouped based on the same object or based on their keys, and placed in the same reducer.
- Final result, at this stage, tourism object information in the form of text is generated, which has produced an object score by applying top-k query calculation on Hadoop MapReduce, capable of determining potential tourism objects that need to be developed based on the object score generated by the system.

D. Testing

In the testing process, Hadoop MapReduce will be installed on each Hadoop Cluster and will execute the distributed program using top-k query calculation for various file sizes and number of nodes used. After the program runs smoothly, analysis will be performed on the execution time on each node that has run various file sizes, so that the response from Hadoop MapReduce can be determined.

IV. Results and Discussion

A. Data Aggregation

To conduct tests on a Hadoop cluster that executes distributed programs using top-k query calculations, the European restaurants data according to TripAdvisor [14] and synthetic data were utilized. The European restaurants data according to TripAdvisor is divided into 5 dimensions and 14 dimensions, each with a data count of 104.855, 209.710, 419.420, 838.840, and 1.048.546. Meanwhile, for synthetic data, it was generated using Java programming language with a data count of 1 million, 4 million, 8 million, 12 million, 16 million, and 20 million with 5 dimensions, and a data count of 10 million with dimensions of 2, 3, 4, and 5

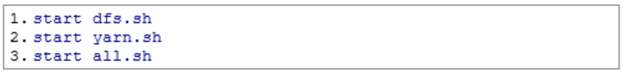
B. Data Normalization

The obtained data from European restaurants data according to TripAdvisor and synthetic data were subjected to data normalization, a process of scaling attribute values using min-max normalization within a range of 0 to 1, where 0 represents the lowest value and 1 represents the highest value. This was done to avoid the domination of larger data on attributes with higher values.

C. System Implementation

The system implementation for data processing in testing a Hadoop cluster that executes distributed programs using top-k query calculations for various file sizes and numbers of nodes involves several stages as follows.

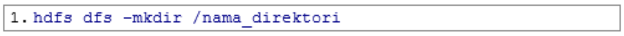
- Building Hadoop Cluster, at this stage, Hadoop Cluster is built using 8 HP PCs with Intel Core i5 CPU 3.00GHz RAM 4GB specifications which have been installed with virtual box for virtual machine purposes with Ubuntu operating system on each PC. Next, Hadoop installation and network configuration are performed so that each PC can connect and form a Hadoop cluster.
- Running Hadoop Cluster, at this stage, the process of running Name Node, Data Node, Resource Manager, and Node Manager is carried out on one of the PCs that become Master Node using the following commands.



```
start dfs.sh
start yarn.sh
start all.sh
```

Fig. 3. Command to run Hadoop cluster

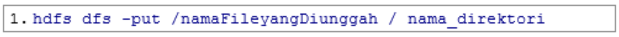
- Creating a directory in HDFS, at this stage, a directory is created in HDFS to store the program file, data used, and output of program execution results run on the cluster using the following command.



```
hdfs dfs -mkdir /nama_direktori
```

Fig. 4. Command to create a directory in HDFS

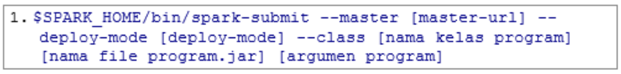
- Uploading program files to HDFS, at this stage, the program and data files used will be uploaded to HDFS so that the program can be run on the Hadoop cluster. For programs using Java programming language, a JAR file is created from the program beforehand. After that, the program and data files are uploaded to HDFS using the following command.



```
hdfs dfs -put /namaFileyangDiunggah / nama_direktori
```

Fig. 5. Command to upload program files to HDFS

- Executing the program, at this stage, the program will be executed 3 times on European restaurants data according to TripAdvisor using 5 dimensions and 14 dimensions, each with a number of data of 104,855, 209,710, 419,420, 838,840, and 1,048,546. In addition, the program is also tested on synthetic data with a number of data of 1 million, 4 million, 8 million, 12 million, 16 million, and 20 million with 5 dimensions, and a number of data of 10 million with 2, 3, 4, and 5 dimensions. Program execution is performed using the command in Source Code 4.4. If the program is successfully executed 3 times and provides results that meet the expected requirements, it will proceed to the next stage. However, if the program cannot run well, program execution will be repeated.



```
$SPARK_HOME/bin/spark-submit --master [master-url] --
 deploy-mode [deploy-mode] --class [nama kelas program]
 [nama file program.jar] [argumen program]
```

Fig. 6. Command to execute the program

- Analysis of results, at this stage, an analysis of the results is carried out by calculating the average execution time from the 3 program execution on each of the testing data that has been run. The results of the analysis will be used to determine the MapReduce performance in applying top-k query calculations to manage data.

D. System Testing Results

D.1. MaterialsTesting European Restaurants Data According to TripAdvisor Using 5 Dimensions

TABLE III. Results of Testing European Restaurants Data According To TripAdvisor Using 5 Dimensions

No.	The Amount of Data	Execution Time (second)	
		Single Node	Multi Node (3 Nodes)
1.	104.855	33	33
2.	209.710	34	34
3.	419.420	40	38
4.	838.840	46	41
5.	1.048.546	50	46

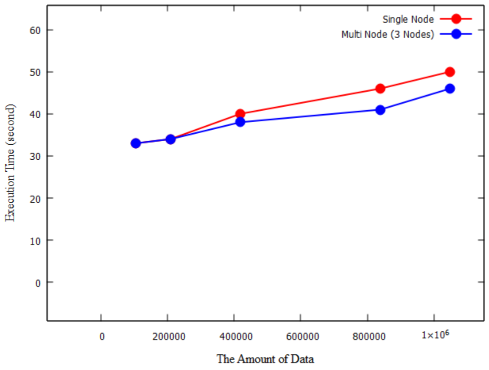


Fig. 7. Graph of the results of testing European restaurants data according to TripAdvisor using 5 dimensions

Based on the results of testing European restaurants data according to TripAdvisor using 5 dimensions presented above, for relatively small data sizes (104,855 and 209,710), the execution time of both single node and multi node with 3 nodes did not differ significantly, taking around 33-34 seconds to complete data processing. However, for larger data sizes (419,420, 838,840, and 1,048,546), the execution time of multi node with 3 nodes was faster than that of single node. This is because in multi node processing, data is processed in a distributed manner across several nodes in the cluster, thus the execution time can be accelerated. For a data size of 419,420, the execution time of multi node with 3 nodes was 38 seconds, while single node took 40 seconds. For a data size of 838,840, multi node with 3 nodes took 41 seconds, while single node took 46 seconds, and for a data size of 1,048,546, multi node with 3 nodes took 46 seconds, while single node took 50 seconds.

D.2. Testing European Restaurants Data According to TripAdvisor Using 14 Dimensions

TABLE IV. Results of Testing European Restaurants Data According To TripAdvisor Using 14 Dimensions

No.	The Amount of Data	Execution Time (second)	
		Single Node	Multi Node (3 Nodes)
1.	104.855	33	37
2.	209.710	36	47
3.	419.420	45	44
4.	838.840	63	49
5.	1.048.546	68	54

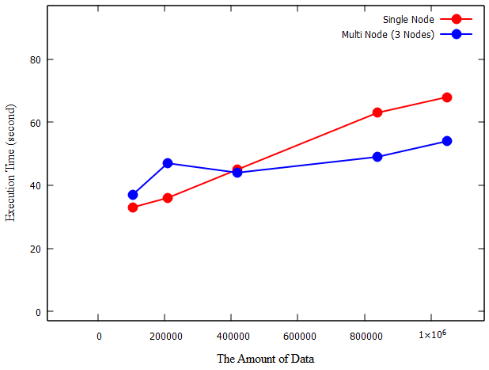


Fig. 8. Graph of the results of testing European restaurants data according to TripAdvisor using 14 dimensions

Based on the above testing results of European restaurants data from TripAdvisor using 14 dimensions, the execution time on a single node tends to significantly increase as the amount of data gets larger. However, on multi node with 3 nodes, the execution time is more stable, but for smaller amounts of data (104,855 and 209,710), the execution time on a single node is faster than on multi node with 3 nodes. This is because on multi node with 3 nodes, data processing is performed in a distributed manner on several machines in the cluster, which requires time for synchronization between nodes or for data transfer between nodes. On the amount of data 104,855, the execution time on a single node is 33 seconds, while on multi node with 3 nodes it is 37 seconds. On the amount of data 209,710, the execution time on a single node increases to 36 seconds, while on multi node with 3 nodes it increases more significantly to 47 seconds. On the amount of data 419,420, the execution time on a single node increases to 45 seconds, while on multi node with 3 nodes it decreases to 44 seconds. On the amount of data 838,840, the execution time on a single node significantly increases to 63 seconds, while on multi node with 3 nodes it remains relatively stable at 49 seconds. On the amount of data 1,048,546, the execution time on a single node increases again to 68 seconds, while on multi node with 3 nodes it increases to 54 seconds.

D.3. Testing Synthetic Data with Amount Data 1 Million, 4 Million, 8 Million, 12 Million, 16 Million, and 20 Million with 5 Dimensions

TABLE V. Results of Testing Synthetic Data with Amount Data 1 Million, 4 Million, 8 Million, 12 Million, 16 Million, and 20 Million with 5 Dimensions

No.	The Amount of Data	Execution Time (second)	
		Single Node	Multi Node (3 Nodes)
1.	1 million	74	75
2.	4 million	157	105
3.	8 million	262	122
4.	12 million	417	175
5.	16 million	499	233
6.	20 million	662	264

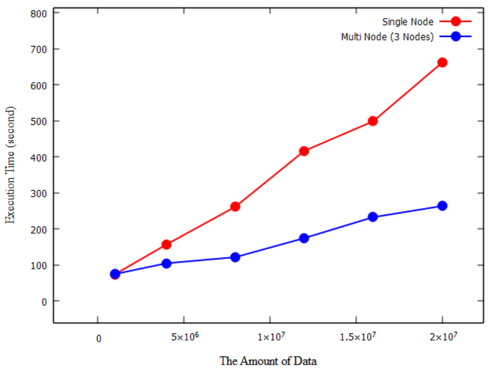


Fig. 9. Graph of the results of testing synthetic data with amount data 1 million, 4 million, 8 million, 12 million, 16 million, and 20 million with 5 dimensions

Based on the results of synthetic data testing with a volume of 1 million, 4 million, 8 million, 12 million, 16 million, and 20 million dimensions 5, on the volume of 1 million data, execution time on single node and multi node with 3 nodes is relatively the same, around 74 seconds and 75 seconds, respectively. However, on larger volumes such as 4 million, there is a significant difference in execution time between single node and multi node. On the volume of 4 million data, execution time on single node is around 157 seconds, while on multi node with 3 nodes it is around 105 seconds. Similarly, on 8 million data, execution time on single node is around 262 seconds, while on multi node with 3 nodes it is around 122 seconds. The difference in execution time between single node and multi node with 3 nodes becomes larger as the volume of data increases, such as on 20 million data, execution time on single node is around 662 seconds while on multi node with 3 nodes it is around 264 seconds. This indicates that execution time on single node increases linearly with the increase in the volume of data, while execution time on multi node with 3 nodes is more stable and even twice as fast in processing data because in multi node data processing can be done in parallel on each node, thus execution time can be accelerated.

D.4. Testing Synthetic Data with 10 Million Data with 2, 3, 4, and 5 Dimensions

TABLE VI. RESULTS OF TESTING SYNTHETIC DATA WITH 10 MILLION DATA WITH 2, 3, 4, AND 5 DIMENSIONS

No.	Number of Dimensions	Execution Time (second)	
		Single Node	Multi Node (3 Nodes)
1.	2 Dimensions	246	113
2.	3 Dimensions	320	132
3.	4 Dimensions	331	156
4.	5 Dimensions	362	167

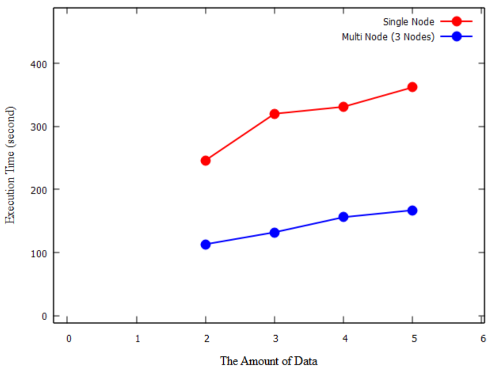


Fig. 10. Graph of the results of testing synthetic data with 10 million data with 2, 3, 4, and 5 dimensions

Based on the results of the synthetic data testing with 10 million data with dimensions 2, 3, 4, and 5 above, the execution time on both single node and multi-node with 3 nodes increased with the increasing dimensions. However, it is observed that multi-node with 3 nodes is faster than single node in all dimensions due to parallel processing of data on several nodes, resulting in faster execution time compared to single node, which only uses one node to process data. In dimension 2, the execution time on single node was 246 seconds, while on multi-node with 3 nodes it was 113 seconds. In dimension 3, the execution time on single node increased to 320 seconds, while on multi-node with 3 nodes it increased to 132 seconds. In dimension 4, the execution time on single node further increased to 331 seconds, while on multi-node with 3 nodes it increased to 156 seconds. In dimension 5, the execution time on single node further increased to 362 seconds, while on multi-node with 3 nodes it increased to 167 seconds.

D.5. Testing Synthetic Data with 20 Million Data with 5 Dimensions Using 2 Nodes, 4 Nodes, 6 Nodes, and 8 Nodes

TABLE VII. RESULTS OF SYNTHETIC DATA WITH 20 MILLION DATA WITH 5 DIMENSIONS USING 2 NODES, 4 NODES, 6 NODES, AND 8 NODES

No.	Number of Nodes	Execution Time (second)
1.	2 *node*	681
2.	4 *node*	349
3.	6 *node*	262
4.	8 *node*	318

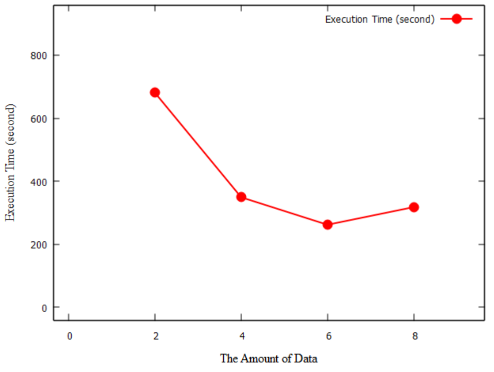


Fig. 11. Graph of the results of testing synthetic data with 20 million data with 5 dimensions using 2 nodes, 4 nodes, 6 nodes, and 8 nodes

Based on the results of the testing using various numbers of nodes with 20 million data with 5 dimensions as described above, the execution time for 2 nodes is the longest, around 681 seconds, while the execution time for 4 and 6 nodes is much shorter, around 349 and 262 seconds, respectively. However, the execution time increases again to around 318 seconds when using 8 nodes. This may be due to the possible overhead in the distribution process, where when the number of nodes is too high, data distribution and synchronization between nodes can take longer than the execution time on only a few nodes. Therefore, the execution time on 8 nodes is longer than on 6 nodes, and processing data with 20 million data points of dimension 5 using 6 nodes is an optimal data processing process because it has the shortest execution time compared to using 2, 4, and 8 nodes.

V. Conclusions and Suggestions

A. Conslusions

Based on the results of the research on the performance of MapReduce in applying top-k query calculations for processing data in culinary tourism potential selection based on restaurant reviews, the following conclusions can be drawn:

- Multi-node with 3 nodes and a large number of dimensions show faster execution time in processing large data (more than 1 million) compared to a single node.
- For data with a size of 20 million and using 5 dimensions, the optimal data processing was achieved using multi-node with 6 nodes as it had the shortest execution time.
- For relatively small data sizes, namely 104,855, 209,710, and 1 million, single node execution time is faster than multi-node with 3 nodes because multi-node with 3 nodes requires time for synchronization in distributed data processing across multiple machines in the cluster.

B. Suggestion

Based on the results of the research, there are several recommendations that can be given, including:

- Conducting testing with larger data sets to determine to what extent the performance of single node and multi node will differ when given larger amounts of data.
- Conducting testing with more nodes to determine the optimal maximum number of nodes in processing data with different sizes and dimensions, so that execution time can be optimized properly.
- Paying attention to the hardware and software specifications used, as well as the proper configuration settings to maximize the performance of the system used.
- Conducting further testing on different types of data and using different data processing methods, in order to broaden insights into the speed and efficiency of data processing.

References

[1] UNWTO, “International Tourism Back to 60% of Pre-Pandemic Levels in January to July 2022,” 26 September 2022, 2022. https://www.unwto.org/news/international-tourism-back-to-60-of-pre-pandemic-levels-in-january-july-2022 (accessed Nov. 04, 2022).

[2] N. H. Ryeng, A. Vlachou, C. Doulkeridis, and K. Nørvåg, “Efficient Distributed Top-k Query Processing with Caching,” Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics), vol. 6588 LNCS, no. PART 2, pp. 280–295, 2011, doi: 10.1007/978-3-642-20152-3_21.

[3] R. Adawiyah and S. Munir, “Analisis dan Evaluasi Algoritma MapReduce wordcount pada Cluster Hadoop Menggunakan Indikator Kecepatan,” J. Inform. Terpadu, vol. 6, no. 1, pp. 14–19, 2020, [Online]. Available: https://journal.nurulfikri.ac.id/index.php/JIT

[4] H. M. Putra, T. Akbar, A. Ahmadi, and M. I. Darmawan, “Analisa Performa Klastering Data Besar pada Hadoop,” Infotek J. Inform. dan Teknol., vol. 4, no. 2, pp. 174–183, 2021, doi: 10.29408/jit.v4i2.3565.

[5] A. L. Ramdani, “Pemilihan Akun Berpengaruh pada Data Twitter Menggunakan Skyline Query dalam MapReduce Framework,” Institut Pertanian Bogor, 2016.

[6] A. M. Ryanto, “Analisis Kinerja Framework Big Data pada Cluster Tervitualisasi: Hadoop MapReduce dab Apache Spark,” 2017.

[7] E. F. Manalu, “Analisis Terhadap Skyline Query dan Top-K Query pada Context Preference Aware Service,” Makal. IF3051, 2009.

[8] D. Rusdiyatno, E. Pramunanto, and A. Zaini, “Sistem Informasi Jalur Alternatif Menggunakan Top-k Query Berbasis WEB Pada BlackBerry TM.” 2012.

[9] H. Wijayanto, W. Wang, W.-S. Ku, and A. L. P. Chen, “LShape Partitioning: Parallel Skyline Query Processing using MapReduce,” IEEE Trans. Knowl. Data Eng., vol. 34, no. 7, 2022, doi: 10.1109/TKDE.2020.3021470.

[10] H. Wijayanto, S. A. Thamrin, and A. L. P. Chen, “Upgrading products based on existing dominant competitors,” Proc. Annu. Hawaii Int. Conf. Syst. Sci., vol. 2020-Janua, pp. 1738–1747, 2021, doi: 10.24251/hicss.2021.211.

[11] I. F. Ilyas, G. Beskales, and M. A. Soliman, “A survey of top-k query processing techniques in relational database systems,” ACM Comput. Surv., vol. 40, no. 4, pp. 1–61, 2008, doi: 10.1145/1391729.1391730.

[12] K. Mouratidis and B. Tang, “Exact processing of uncertain topk queries in multicriteria settings,” Proc. VLDB Endow., vol. 11, no. 8, pp. 866–879, 2018, doi: 10.14778/3204028.3204031.

[13] D. A. Nasution, H. H. Khotimah, and N. Chamidah, “Perbandingan Normalisasi Data Untuk Klasifikasi Wine Menggunakan Algoritma K-NN,” CESS (Journal Comput. Eng. Syst. Sci., vol. 4, no. 1, pp. 78–82, 2019.

[14] S. Leone, “TripAdvisor European restaurants,” 2021. https://www.kaggle.com/datasets/stefanoleone992/tripadvisor-european-restaurants?resource=download