# The Performance of Hadoop Cluster MapReduce on Private Cloud Computing for Skyline Query Computations

Annisa Octavyanti Hakim [1], Heri Wijayanto[1], I Gde Putu Wirarama[1]
[1] Dept Informatics Engineering, Mataram University Jl.
Majapahit 62, Mataram, Lombok NTB, INDONESIA
*Email:* annisaoctavya@gmail.com, [heri, wirarama]@unram.ac.id

***The use of Hadoop MapReduce technology requires procuring infrastructure at a significant cost, especially when the scale of processing increases. To maximize the utility of Hadoop processing, cloud computing offers easy-to-use infrastructure for big data processing with a combination of private cloud services and Infrastructure as a Service (IaaS). In this thesis, the author characterizes and assesses the execution performance of big data on Hadoop MapReduce virtual cluster instances built using the University of Mataram's private cloud. Using the Skyline Query algorithm, clusters will be tested by varying the amount of data, the number of machines and the HDFS blocksize variations on 3 types of synthetic data, namely anti-correlated, correlated and independent. The author also tracks the performance of Hadoop private cloud instances using running time parameters and compares them based on the same tests with Hadoop clusters of physical infrastructure. Test results on private cloud clusters show that increasing the amount of data from 1.5 million to 12 million using 4 machines causes an increase in completion time for anti-correlation data (168%), correlation data (194%) and independent data (126%). The same performance trend is also experienced in physical Hadoop clusters. In the second scenario, the private cloud cluster shows that increasing the number of machines from 1 machine to 7 machines causes cluster performance to increase until it reaches ideal conditions. While on the physical Hadoop cluster, scaling machines to 7 machines causes communication overhead between nodes. In processing data with a block size of 512 MB using 12 million data or 1.06 GB and 7 machines, this is the most optimal HDFS block size in this study because it has the shortest execution time. Based on the t statistical test using the average processing time, it was concluded that the Hadoop cluster virtualized on a Private Cloud with the Intel(R) Xeon (R) E3-1225 v5 @ 3.30 GHz RAM 16 GB engine specifications, works much better in executing applications Skyline compared Hadoop cluster built on a physical machine with machine specs Intel Core i5 CPU @ 3.00GHz 4GB RAM.***

***Key words*: Hadoop MapReduce, Cloud Computing, Private Cloud, DistributedSkyline Query, MR-BNL, big data.**

## I. INTRODUCTION

Internet technology has developed very rapidly since its emergence in the 1960s [1]. Internet users who continue to increase along with increasingly diverse application services are triggers for data consumption to continue to increase non-stop. Large amounts of data without proper management mechanisms will only become passive objects that cannot be utilized anymore. Therefore, a capable architecture is needed to manage big data [2].

One of the most popular big data frameworks today is Hadoop. Hadoop is an architecture that uses a distributed parallel concept to process large volumes of data using the MapReduce programming model through a group of computers connected to each other via a network (cluster) [3]. MapReduce divides data into many shards where each shard will be processed at each node in a cluster.

At this time, managing big data using Hadoop has its own challenges in terms of providing, setting and maintaining large-scale infrastructure. Initial investment costs are required in terms of infrastructure, operations, IT experts and ongoing maintenance which is certainly not small. This makes Hadoop implementations with limited physical machines possible. To solve this challenge, cloud computing offers the concept of processing computing resources through the internet network (cloud) at a cost as large as that used by the user. This can help users to focus more on their work instead of worrying about the availability of IT infrastructure, resources and experts.

Computing resources offered by cloud computing such as servers, storage, software and networks are provided in the Infrastructure as a Service (IaaS) layer. The IaaS layer will work using virtualization principles that allow users to choose and use the type and configuration of infrastructure needed and reduce or enlarge the scale of services or computing resources used.

In cloud computing, there are several deployment models, one of which is the private cloud. This service is widely used by users such as companies and universities who want exclusive control. Private cloud gives full control to its users by providing special access to the network and infrastructure that can be customized. Private cloud implementation with Infrastructure as a Service (IaaS) will be provided in the form of virtual instances or virtual infrastructure that can be requested according to internal needs. This virtual infrastructure works like a machine with components of storage, RAM, disk space, operating system, network and CPU processing power [4]. IaaS with virtual machines can be managed flexibly and can technically replace physical servers, data center resources, network tools and other physical components [5].

Cloud computing private cloud has become an excellent solution to improve reliability, have high performance and reduce computing costs. This statement is in line with [6], which examined performance comparisons between private cloud clusters for High Performance Computations (HPC) and Hadoop clusters from physical machines. The result is that virtual private cloud clusters (KVM and VMware ESXi) have better performance than physical infrastructure clusters.

The advantages of this private cloud can be utilized to overcome the limitations of providing physical machines with ideal specifications for complex scale big data computing in the Informatics Engineering study program laboratory, Mataram University. Currently, the laboratory of 2 Informatics Engineering study program has 21 PCs with Intel(R) Core (TM) i5-9500 CPU specifications @ 3.00GHz 3.00 GHz, 4.00 GB RAM (3.78 GB usable) and 500 SSD memory GB. Based on [7], these specifications are the minimum specifications for Hadoop computing. So, for better Hadoop computing, virtualization technology with private cloud will be very well implemented.

The focus of this research lies in the performance of the Hadoop private cloud cluster which will be built on a server

owned by the Univeritas Mataram, in completing the Hadoop MapReduce computation. The Hadoop ecosystem that was built will be used to execute large amounts of data with several test scenarios, one of which is the Skyline Query algorithm to assess Hadoop cluster performance. Skyline Query is a search method for a set of important objects that have better criteria than other objects in the data set. This algorithm was chosen because the complexity of this algorithm is very dependent on the number of dimensions and the size of the dataset used [8]. As a comparison, similar requirements will also be executed on a physical machine. Both of these implementations will be compared for their performance in terms of execution speed or running time when running computations.

## II. Literature Review

### *A. Related Research*

In research conducted by Quidad Achahbar entitled "The Impact of Virtualization on High Performance Computing Clustering in the Cloud" in 2014, regarding the performance evaluation of one of the cloud computing services namely HPCaaS or High Performance Computing as a Service using MapReduce and different virtualization techniques. HPC is built on a Private Cloud using Openstack. In this study, three experiments were carried out on 3 different clusters, namely Hadoop Physical Cluster (HPhC), Hadoop Virtualized Cluster using KVM (HVC-KVM) and Hadoop Virtualized Cluster using VMware ESXi. Furthermore, to determine the impact of implementing machine virtualization technology, performance testing was carried out using 2 benchmarks, namely Terasort and TestDFSIO. In Terrasort the data sizes to use are 100 MB, 1GB, 10 GB and 30 GB and 100 MB, 1GB, 10 GB and 100 GB for TestDFSIO. From the experimental results, it was found that the virtual cluster performs computationally better than the physical cluster when processing and handling HPC, especially when there is little overhead on the virtual cluster. In addition, based on testing it was found that Hadoop VMware ESXi clusters perform better in sorting large data sets (more calculations), and Hadoop KVM clusters perform better on I/O operations [6].

In the research conducted by Vladimir Starostenkov and Kirill Grigorchuk entitled "*Hadoop Distributions: Evaluating Cloudera, Hortonworks, and MapR in Micro-benchmarks and Real-world Applications*" in 2011, compared three open-source Hadoop distributions namely Cloudera, Hortonworks Data Platform and MapR using Micro-benchmarks. Test parameters are based on CPU, disk, RAM, network, and JVM parameters with Ganglia Monitoring. All three distributions are founded on the ProfitBricks cloud computing platform with each node having four CPU cores, 16 GB RAM and 100 GB virtual disk space and cluster sizes ranging from 4 to 16 nodes. Based on the test results, it can be concluded that the type of Hadoop distribution has a much smaller impact on the overall system throughput than the MapReduce configuration parameters. In addition, using cloud computing as a distribution platform for Hadoop allows users to scale horizontally and vertically. So it is important for users to choose an IaaS platform that gives freedom in configuring the infrastructure [9].

In research conducted by Md. Anisuzzaman Siddique, et al untitled "*MapReduce Algorithm for Variants of Skyline Queries: Skyband and Dominating Queries*", discuss ways to speed up and improve Skyline computational efficiency using the MapReduce framework. Based on the MapReduce framework, there are three Skyline query variants discussed, namely MR-BNL, MR-SFS and MR-Bitmap. Furthermore, these three algorithms were evaluated and compared using a series of different experiments including data distribution settings, dimensions, buffer size and cluster size. Experimental results show that MR-BNL and MR-SFS are good in many use cases, but still suffer from disadvantages in terms of parallel processing dimensions. While MR-Bitmap works quite well when the bitmap can fit into the memory of a node [10].

Therefore, the difference between this research and previous related research is that this research will analyze the performance of the Hadoop MapReduce distributed system built on a private cloud, then compare it with the Hadoop cluster using a physical engine to find out how fast the performance of the two is in executing the Skyline Query MR-BNL application. System performance will be tested based on processing time using several types of synthetic data that are generated into several file sizes, number of nodes or machines and HDFS block sizes.

### *B.Supporting Theory*

#### *B.1. Cloud Computing*

Cloud computing refers to a computing model that provides easy access and rapid provisioning on demand by users to shared computing resources (network, servers, storage, applications, and services) over a network with minimal management effort and service provider interaction. The National Institute of Standards and Technology (NIST) defines cloud computing in five characteristics with three service models and five types of deployment as shown in Figure 1.

#### *B.2. Private Cloud*

Exclusive use of cloud infrastructure for one organization with multiple users in it. This type of deployment allows users within an organization or third parties to own, manage and operate the cloud infrastructure on premise or off premise. Examples of private clouds are Proxmox VE, AWS, Azure and many more [11].

#### *B.3. Hadoop MapReduce*

MapReduce works by dividing the process into two main phases, namely map and reduce. In general, MapReduce will process data in four stages, namely mapping, shuffling, merging and reducing. Each phase produces keys and values as input and output [3], [12], [13]. Following are the stages in MapReduce:

- Mapping. The map stage or function will receive and process the input by breaking the large input into smaller sizes which can then be distributed randomly in the same amount to the processing NameNode in the map function [13]. The output of this process is in the form of data pairs consisting of keys and values. The results of this processing will be taken by the reducer and will enter the shuffling stage.
- Shuffling. This stage is an intermediate stage because it is between the map and reduce processes. In this process data retrieval occurs from the mapping process by reduce. Master has distributed different K keys on each map, as well as reduce. Each key on reduces will then be matched on each map. If the key owned by reduces is the same as the one on the maps, then a data pair (Kr, Vr) is formed which will be retrieved by reduces.
- Merging. This stage is the process of combining data according to the given key to form a data pair (Kr, Vr) that has gone through the shuffling process.
- Reducing. This stage is the final stage to get the complete final output. The process of merging all data on reduces, analyzes and sorts data according to the given reduce function is carried out.

The diagram below is the workflow of the MapReduce process.

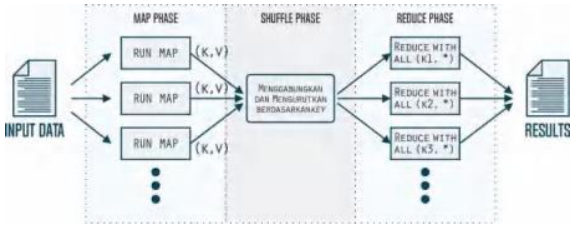


Fig. 1. Mapreduce workflow [3]

*B.4. Hadoop Cluster*

A Hadoop cluster is a group of computers known as nodes that are connected to each other via a network to jointly perform parallel computations on a large dataset [14]. In a Hadoop MapReduce cluster, it consists of master and slave nodes that are connected to each other. The master node refers to the computer that is responsible for managing the process of distributing data between nodes in the cluster, while the slave node is responsible for carrying out the MapReduce process and storing data blocks [15]. Master nodes usually use higher quality hardware than slave nodes. In addition, the master node usually includes a NameNode, Secondary NameNode and ResourceManager inside, where each of these components will run on a separate machine. Meanwhile, slave nodes usually consist of virtual machines that run DataNode and NodeManager services based on instructions from the master node.

*B.5. Skyline Query*

The Skyline Algorithm is a data point search method that is not dominated by other data points. Data points can be represented by a tuple against a certain criterion or attribute. This algorithm will produce a number of superior data on all criteria or only on one particular criterion [16].

For example, [17] the selection of the best hotel is based on two attributes, namely price and distance to a place the customer wants to visit such as tourist attractions, beaches and so on. Based on these two attributes, we can retrieve the Skyline object which can help customers determine the best hotel. In general, customers will look for hotels with a combination of lower prices and closer proximity. An example of this Skyline is shown in Table I and Figure 2.

TABLE I. DATA HOTEL DENGAN ATRIBUTNYA

| ID | Price | Distance |
|---|---|---|
| $h_1$ | 3 | 8 |
| $h_2$ | 5 | 4 |
| $h_3$ | 4 | 3 |
| $h_4$ | 9 | 2 |
| $h_6$ | 7 | 3 |

In Table I, there are 5 hotel data objects with price and distance information. Based on the attributes in Figure 2.6, {h1, h¬3, h4} is a Skyline object because it has a smaller price and distance compared to other objects. With this it is said that the objects {h1, h¬3, h4} do not dominate each other. While the object {h2, h5} is dominated by h¬3. Skyline objects are skyline objects that are not dominated by other objects, in this case the skyline objects are hotel sets {h1, h3, h4}.

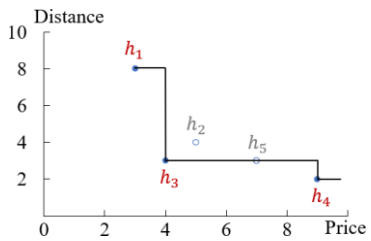


Fig. 2. Skyline case example

*B.6. MR-BNL*

Block Nested Loops (BNL) is a variant of the Skyline algorithm that performs computations with loops to read a record repeatedly. At each data reading, the window will save the Skyline candidate in main memory. When a record is read, the record (eg record *p*) will be compared with all the tuples in the window. Then, there will be three possibilities: the p record will be trimmed because it does not dominate any data point in the window, or the p record will be inserted into the window, or the data point in the window that is dominated by p will be deleted. With N data size and D dimensions, the performance speed of BNL is greatly influenced by the window size and the degree of order of the original data [18].

## III. RESEARCH METHODOLOGY

*A. Tools and Materials*

*A.1 Hadoop Cluster*

In this study, the Hadoop cluster was built on Infrastructure as a Service services on Private Cloud using a Data Communication and Embedded System 2 laboratory server located at UPT PUSTIK Universitas Mataram. The cluster to be launched is in the form of a virtual instance or virtual machine of 1 node as initial initialization. Each node will be installed with Linux Ubuntu 22.04.1 LTS as the basis for the Hadoop cluster operating system.

The Hadoop cluster distribution in the form of a virtual machine will be created using the PUSTIK Universitas Mataram server computer with the following specifications:

- Processor : Intel(R) Xeon (R) E3-1225 v5, 4 Cores, CPU @3.30 GHz
- RAM : 16GB
- Memory : HDD, 1TB

As a comparison, Hadoop configuration was performed with similar requirements on a virtual cluster using a physical computer at Laboratory 2 of Data Communication and Embedded Systems Informatics Study Program. This computer cluster will be given the same treatment as a virtual cluster. The Hadoop Cluster architecture to be built is shown in Figure 3.

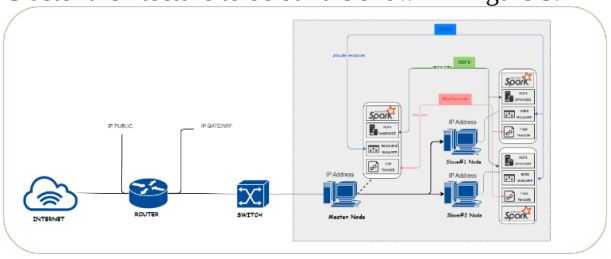


Fig. 3. Hadoop cluster architecture

*A.2 Software*

- Ubuntu 22.04.1 LTS
- Apache Hadoop
- Apache Spark
- Java Programming Language
- Java Development Kit (Open JDK)

*A.3 Data*

In data distribution, the authors use various types of synthetic data, namely correlated, uncorrelated and independent, each with a size of 1 GB. These data are [30]:

- Independent: all attribute values are created independently using a uniform distribution.
- Correlated: refers to data that is correlated or connected to one another.
- Uncorrelated: data that are not related to each other. In a sense, the data that is owned is only good in one dimension, not in other dimensions.

## B. Research Flow

The research flow contains the steps that the researcher will take in solving the main issues raised in this study. The research flowchart can be seen in Figure 4.

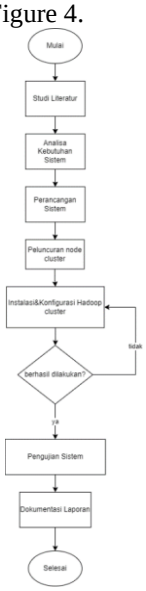


Fig. 4. Research Flow

- Literature review, this stage is carried out to collect information.
- Analysis of system requirements, at this stage the researcher will collect system requirements such as hardware, software, and data that needs to be processed by the system.
- System design. At this stage, a test scenario is designed as the main focus to be studied.
- Virtual and physical clusters will be launched according to the needs that have been previously designed.
- Installation and configuration process on each cluster node, this stage includes the process of preparing hardware and software such as installing the operating system and installing other software.
- In the testing phase, an evaluation will be carried out on the Hadoop cluster that has been built with a private cloud using several scenarios that have been designed.
- Then the test results will be included in the report documentation as accountability material.

## C. Testing

In this study, because the research focus is on testing Hadoop cluster performance using IaaS Private Cloud services, it is necessary to develop a test scenario design. Testing will be carried out using several cases as shown in Table II.

TABLE II. CLUSTER PERFORMANCE TESTING SCENARIO DESIGN

| Case | | Testing Steps | Expected Result |
|---|---|---|---|
| Skyline Computing with MR-BNL | 1. | Input data | The system is capable of generating local and global skylines |
| | 2. | Splitting the data equally by $2^d$ on the mapper engine | |
| | 3. | Provide the *d-bit* flag | |
| | 4. | Local skyline scanning with MR-BNL | |
| | 5. | Local skyline combined using flags | |
| | 6. | Reducing with MR-BNL | |
| | 7. | Global skyline is generated | |
| File size variations | 1. | Of the total dataset size owned, execution is carried out periodically with the first file size of around 100 MB or a total of 1.5 million data. | The speed of execution time gets slower as the file size increases |
| | 2. | The file size will continue to increase until the data is around 1 GB in size with fractions of 200 MB (2.5 million), 400 MB (5 million), 800 MB (10 million) to 1 GB (12 million). | |
| Number of machines variations | 1. | Hadoop MapReduce runs with 1 node using a certain file size. | Hadoop MapReduce's computational speed increases as the number of machines scales |
| | 2. | Furthermore, machines are scaled from 2 engines to 7 engines. | |
| HDFS Block sizes variations | 1. | Run Hadoop MapReduce computation with Blocksize HDFS which is smaller than default (128 MB) which is 64 MB | There is an increase in execution time as the HDFS block size increases. |
| | 2. | Added block size to 128MB, 256MB, and 512MB | |

According to Table II, in assessing cluster responses, the authors prepared several test scenarios which were carried out in stages. Each test is carried out using the MR-BNL algorithm with an initialized cluster to look for Skylines in large data. In the Skyline Block Nested Loop (BNL), the MapReduce process to produce a local skyline and a global skyline consists of two phases. The first phase is the distribution of data partitions to the mapper and the second phase is computing the local skyline on each partition with BNL to produce a global skyline.

In the first phase, the data will be divided into chunks of $2^d$ based on the median of each dimension. Each dimension will be cut into two parts where the high part contains the larger value and the low part contains the smaller value. Then a flag or marker is given for the part that has a higher 1 and a lower 0. So that identification will be done with a *d-bit* mark. On each partition, a skyline object will be searched by scanning, and the skyline candidate will be saved in the window or buffer memory. Every time a new data point is read, the data point (example:*P*) will be compared with all the other tuples in the window one by one. If *P* dominates one or more data points in the window, then those data points will be deleted, while *P* will be entered into the window. Then in the second phase, before entering the reducer, the local skyline will be combined using flags to reduce unnecessary comparisons. In this phase, the computation will be performed again using MR-BNL. Skyline computation using the MR-BNL algorithm will be shown in Figure 5.

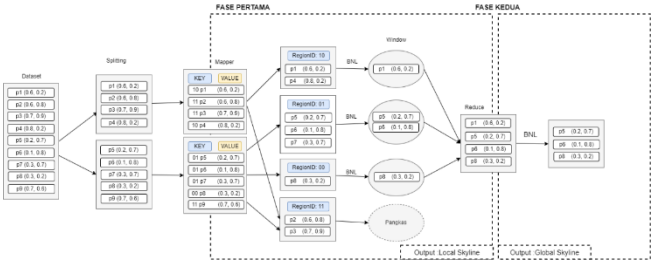


Fig. 5 Skyline with MR-BNL

# V. RESULTS AND DISCUSSION

## A. Cluster Performance Test Results

This section presents the findings obtained after running each experiment. The performance results of the Hadoop MapReduce cluster are shown after running Skyline MR-BNL computations under various conditions on a private cloud and on a physical computer. Clusters were tested with a variety of conditions such as changes in file size, differences in the number of machines to variations in HDFS data block sizes. The authors have used three types of synthetic data input and used the same configuration parameters for realistic comparisons. For each test scenario, the running time of jobs is written in seconds based on three test attempts. Graphs for each scenario on physical clusters and virtual private cloud clusters are plotted to visualize computing

performance.

TABLE III. CLUSTER SPESIFICATIONS

| Klaster Hadoop Private Cloud | | | | | Klaster Hadoop Fisik | | | | |
|---|---|---|---|---|---|---|---|---|---|
| Hostname | Spesifikasi | | | | Hostname | Spesifikasi | | | |
| | Processor | Core(s) | RAM | SSD | | Processor | Core(s) | RAM | SSD |
| Master | Intel(R) Xeon (R) E3-1225 v5 @ 3.30 GHz | 2 | 2.7 GB | 35 GB | Master | Intel Core i5 CPU @ 3.00GHz | 4 | 2.7 GB | 80 GB |
| Slave1 | Intel(R) Xeon (R) E3-1225 v5 @ 3.30 GHz | 1 | 2 GB | 25 GB | Slave1 | Intel Core i5 CPU @ 3.00GHz | 3 | 2 GB | 50 GB |
| Slave2 | Intel(R) Xeon (R) E3-1225 v5 @ 3.30 GHz | 1 | 2 GB | 25 GB | Slave2 | Intel Core i5 CPU @ 3.00GHz | 3 | 2 GB | 50 GB |
| Slave3 | Intel(R) Xeon (R) E3-1225 v5 @ 3.30 GHz | 1 | 2 GB | 25 GB | Slave3 | Intel Core i5 CPU @ 3.00GHz | 3 | 2 GB | 50 GB |
| Slave4 | Intel(R) Xeon (R) E3-1225 v5 @ 3.30 GHz | 1 | 2 GB | 25 GB | Slave4 | Intel Core i5 CPU @ 3.00GHz | 3 | 2 GB | 50 GB |
| Slave5 | Intel(R) Xeon (R) E3-1225 v5 @ 3.30 GHz | 1 | 2 GB | 25 GB | Slave5 | Intel Core i5 CPU @ 3.00GHz | 3 | 2 GB | 50 GB |
| Slave6 | Intel(R) Xeon (R) E3-1225 v5 @ 3.30 GHz | 1 | 2 GB | 25 GB | Slave6 | Intel Core i5 CPU @ 3.00GHz | 3 | 2 GB | 50 GB |

### *A.1 Results and Analysis of Hadoop MapReduce – Private Cloud Virtual Cluster Testing Results*

#### *A.1.1 Scenario of Variation File Sizes*

Running a wide variety of data sizes with 4 machines doesn't take long. The engine takes a different average time for each dataset type and size. Of the three datasets, anti-correlated synthetic data tends to provide higher computation time than the others. This is because the anti-correlated data generates more skyline points than the other two datasets, so the global skyline search time results in a longer time. This can be seen in Table IV.

TABLE IV. COMPUTATION RESULTS OF SCENARIO 1 – HADOOP PRIVATE CLOUD CLUSTER

| Total Data (Million) | Dataset type | | |
|---|---|---|---|
| | Correlated | Anticorrelated | Independent |
| 1.5 | 35 | 51 | 47 |
| 2.5 | 46 | 56 | 52 |
| 5 | 50 | 63 | 53 |
| 8 | 62 | 101 | 73 |
| 10 | 70 | 129 | 78 |
| 12 | 103 | 137 | 106 |

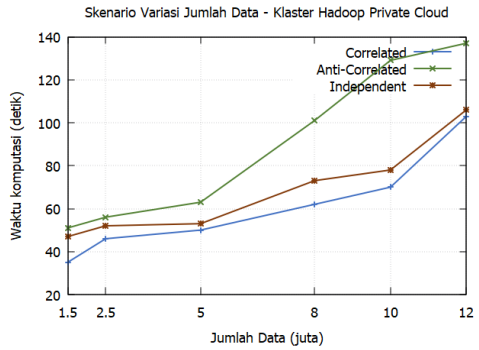


Fig. 6 Computation Results of Scenario 1 – Hadoop Private Cloud Cluster

Based on Figure 6, in general the three datasets show an increase in computation time as the volume of data increases. The increase in data volume resulted in more and more skyline candidates that had to be compared one by one to find a global skyline using the Block Nested Loops algorithm, so the execution time was getting longer.

#### *A.1.2 Scenario of Variation Number of Machines*

This test scenario displays the level of significance of changes in cluster performance partially from the execution time before and after experiencing a change in the number of nodes. In this scenario, the dependent variable is set, namely the amount of data measuring 12 million, while the independent variable is the number of virtual machines from 1 machine to 7 machines. The selection of this amount of data was made after observing the increase in cluster performance in the three datasets in the previous scenario. Table V shows the number of machines giving a generally significant performance increase in executing the three types of datasets when the cluster is increased from 1 machine to 7 machines.

TABLE V. COMPUTATION RESULTS OF SCENARIO 2 – HADOOP PRIVATE CLOUD CLUSTER

| Number of machines | *Dataset types* | | |
|---|---|---|---|
| | *Correlated* | *Anticorrelated* | *Independent* |
| 1 | 110 | 128 | 119 |
| 2 | 104 | 116 | 116 |
| 3 | 104 | 113 | 110 |
| 4 | 97 | 111 | 109 |
| 5 | 94 | 109 | 108 |
| 6 | 93 | 106 | 106 |
| 7 | 87 | 100 | 91 |

Figure 7 clearly illustrates the benefits of scaling a cluster. This can be seen from the fact that the average computing time for Skyline decreased as the number of machines increased. For example, running 12 million or 1.06 GB of anti-correlated data using 1 node takes about 128 seconds, while using 7 nodes it takes only 100 seconds or 28 seconds less (compute time decreased by 21.8%). Almost similar to anti-correlated data, correlated data also shows a decrease in computation time each time the number of machines is added. However, when the machine is scaled from 2 machines to 3 machines, the computation time runs constant. This is presumably due to network bottlenecks which are a common problem for Hadoop computing. Furthermore, a computational reduction of 16.3% was obtained when the application was run from 3 machines to 7 machines. When running 1.06 GB of independent data, the range of computation time required is similar to anti-correlated data. The average data computing time using 1 machine with 7 machines decreased to 28 seconds or 23.5%.

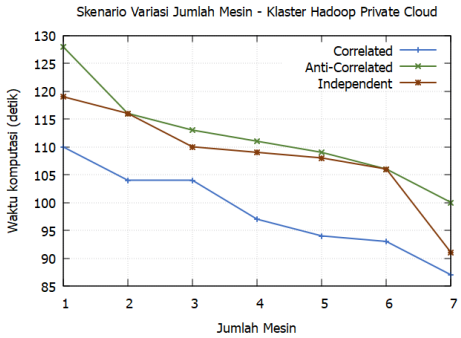


Fig. 7 Computation Results of Scenario 2 – Hadoop Private Cloud Cluster

#### *A.1.3 Scenario of Variation HDFS Data Block Sizes*

Figure 8 shows that block size can affect the speed performance of Hadoop MapReduce in executing the Skyline MR-BNL application for each trial using a 1.06 GB file at various block sizes. In the third scenario, the third dataset with a total of 12 million or 1.06 GB of data will be cut into several blocks according to block sizes of 64 MB, 128 MB, 256 MB and 512 MB. An overview of the block pieces can be seen in Figure 4.39. From the figure it can be seen that the file size of 1.06 GB with the default block (128 MB) is cut into 9 blocks with 3 replications which will be stored on each node.

TABLE VI. COMPUTATION RESULTS OF SCENARIO 3 – HADOOP PRIVATE CLOUD CLUSTER

| Block Sizes | *Dataset types* | | |
|---|---|---|---|
| | *Correlated* | *Anticorrelated* | *Independent* |
| 64 *MB* | 103 | 118 | 113 |
| 128 *MB* | 99 | 106 | 100 |
| 256 *MB* | 91 | 105 | 93 |
| 512 *MB* | 80 | 91 | 89 |

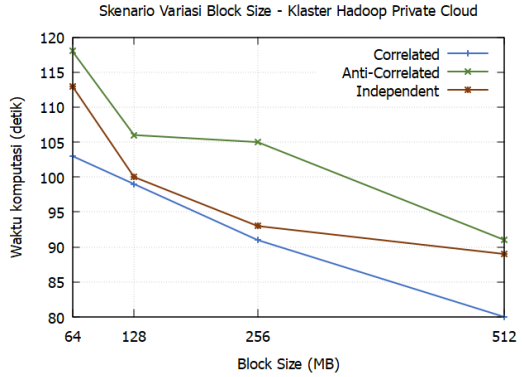


Fig. 8 Computation Results of Scenario 3 – Hadoop Private Cloud Cluster

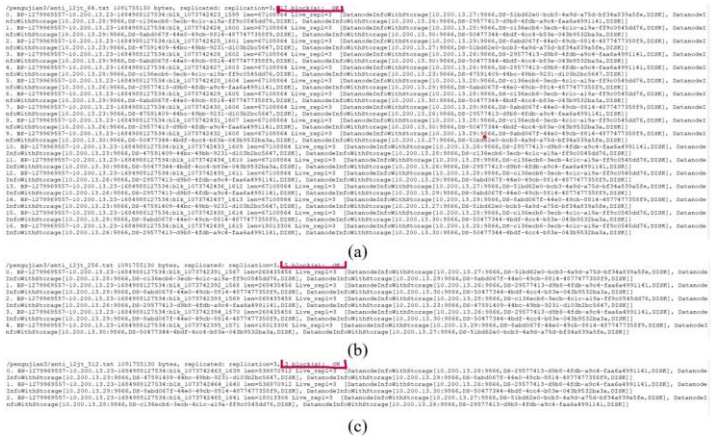

Fig. 9 Block chunks in a 1.06 GB file anti-correlated with (a) 64 MB block size (b) 256 MB block size (c) 512 MB block size

In Figure 9 (a), Figure 9 (b) and Figure 9 (b) shows that the number of blocks at a file size of 1.06 GB with a block size of 64 MB produces 17 blocks, more than the block sizes of 256 MB and 512 MB, respectively each produces 5 blocks and 3 blocks. The smaller number of blocks will reduce the metadata size of the namenode there by speeding up the work process of the namenode. In addition, the number of blocks in HDFS Hadoop determines the number of tasks that must be done by MapReduce. A small number of blocks means a small number of tasks. A small number of tasks can make it easier for the MapReduce task scheduler to schedule a given task thereby reducing the work of the MapReduce scheduler task. A small number of tasks can also reduce communication time for task requests between the MapReduce task scheduler and ResourceManager and ResourceManager and NodeManager. This of course will have an impact on the computing speed of the Hadoop MapReduce that is running.

Figure 8, in outline shows that adding block sizes to the three types of datasets can speed up the MapReduce process in Hadoop. When using a block size of 64 MB with 17 block chunks, Hadoop MapReduce computation runs the slowest compared to using a block size of 128 MB, 256 MB and 512 MB. Meanwhile, the fastest computation time is shown when the block size is 512 MB with the number of blocks produced only 3 blocks.

### *A.2 Results and Analysis of Hadoop MapReduce Physical Cluster Testing Results*

#### A.2.1 *Scenario of Variation File Sizes*

Running the MR-BNL skyline computation using physical clusters shows that it takes more time to generate skyline points for data sizes of 10 GB and 12 GB. Using 4 slave nodes, the results of testing 3 types of synthetic data show the effect of the amount of data on Hadoop MapReduce computation time. Table VII shows the results of skyline computation using three types of synthetic data and is conceptualized in Figure 12.

TABLE VII. COMPUTATION RESULTS OF SCENARIO 1 – HADOOP PHYSICAL CLUSTER

| Total Data (Million) | Dataset type | | |
|---|---|---|---|
| | Correlated | Anticorrelated | Independent |
| 1.5 | 65 | 114 | 68 |
| 2.5 | 86 | 170 | 86 |
| 5 | 87 | 142 | 90 |
| 8 | 118 | 145 | 140 |
| 10 | 132 | 349 | 146 |
| 12 | 135 | 481 | 156 |

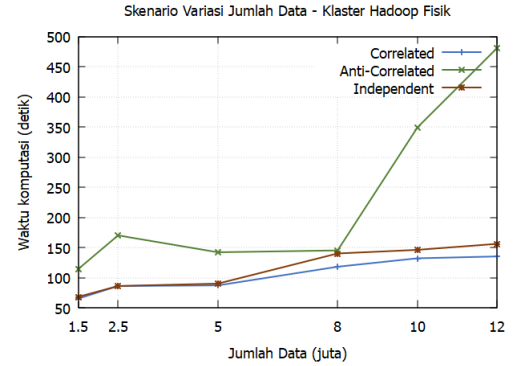


Fig. 10 Computation Results of Scenario 1 – Hadoop Physical Cluster

Figure 10 illustrates the effect of increasing the amount of data which in general can reduce cluster performance as the amount of data increases. In accordance with the number of skyline points generated in the three datasets, the anti-correlated data with the most skyline points certainly requires the highest computation time compared to other data. Meanwhile, independent and correlated data require not much different time to process data.

#### A.2.2 *Scenario of Variation Number of Machines*

The execution time graph of the skyline query computation with variations in the amount of data becomes a benchmark in determining the amount of data to be processed in the next scenario. This test scenario displays the level of significance of changes in cluster performance partially from the execution time before and after experiencing a change in the number of nodes. In the physical Hadoop cluster, the dependent and independent variables are set the same as testing the variation in the number of machines in the Hadoop Private Cloud cluster. Table 4.6 shows the number of machines that have a large influence on data execution time totaling 12 million.

TABLE VIII. COMPUTATION RESULTS OF SCENARIO 2 – HADOOP PHYSICAL CLUSTER

| Number of machines | *Dataset types* | | |
|---|---|---|---|
| | *Correlated* | *Anticorrelated* | *Independent* |
| 1 | 171 | 201 | 179 |
| 2 | 154 | 188 | 154 |
| 3 | 148 | 160 | 146 |
| 4 | 139 | 156 | 145 |
| 5 | 134 | 154 | 141 |
| 6 | 167 | 148 | 141 |
| 7 | 205 | 250 | 225 |

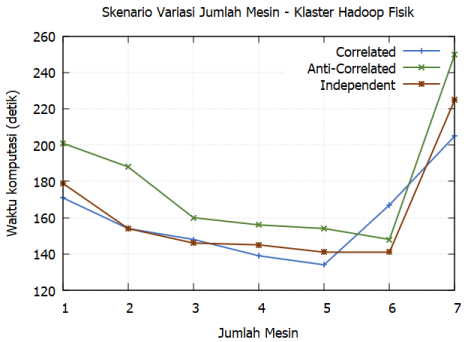


Fig. 11 Computation Results of Scenario 2 – Hadoop Physical Cluster

Running all three synthetic data over a physical cluster yields different results. Generally, an increase in the number of machines will speed up the running of the program. However, running multiple machines at the same time is vulnerable to overhead conditions which increase computation time. In Figure 4.42, the cluster runs optimally for all types of datasets up to 5 nodes. However, when the machines were added to 6 machines, different responses were found for each type of dataset. In general, the cluster shows non-optimal performance when

running the three datasets using 7 nodes. This increase in computing time is thought to be triggered by the complexity of the process of data distribution, synchronization between nodes and communication between Hadoop daemons when the number of machines is added as well as network bottlenecks. Overhead causes excessive computation time and the cluster does not work optimally. There was an average increase in computation time of 50.3% for the three datasets. Excessive computation time due to overhead between nodes also occurs in research [19],[6]. In [6], overhead occurs when processing data of 100 MB, 1 GB, 10 GB and 100 GB using 7 machines and 8 machines for TestDFSIO-Read Performance. In addition, 8 VMware ESXi virtual nodes experienced an overhead condition which was allegedly caused by excess memory, high latency levels, and resource shortages when executing a 30 GB Terrasort. According to [20], the overhead on virtual machines ranges from 2-10% depending on the type of application. However, there are also cases where a virtualized Hadoop cluster has better computation than a physical Hadoop cluster due to better resources. One of the cases is this research. Virtual Hadoop clusters are built with better machine resource specifications compared to physical Hadoop clusters, so they have better performance. The specifications for the two clusters can be seen in Table III.

A.2.3 *Scenario of Variation HDFS Data Block Sizes*

TABLE IX COMPUTATION RESULTS OF SCENARIO 2 – HADOOP PHYSICAL CLUSTER

| Block Sizes | *Dataset types* | | |
|---|---|---|---|
| | *Correlated* | *Anticorrelated* | *Independent* |
| 64 *MB* | 133 | 160 | 137 |
| 128 *MB* | 130 | 157 | 134 |
| 256 *MB* | 128 | 132 | 129 |
| 512 *MB* | 116 | 127 | 121 |

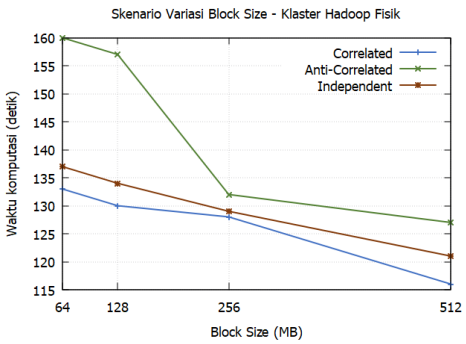


Fig. 12 Computation Results of Scenario 3 – Hadoop Physical Cluster

The graph in Figure 12 shows that the execution time of the Skyline MR-BNL application on a file size of 1.06 GB as the block size increases decreases in all three datasets, including anti-correlated data which decreases by 20.6% from 160 seconds in a 64 MB block to 127 seconds on 512MB blocks. A decrease in execution time was also experienced on correlated data of 12.7% from 133 seconds on 64 MB to 116 seconds on 512 MB. The decrease in Hadoop MapReduce execution time on the three datasets tested shows that the large block size affects the computation process and the larger the HDFS block size, the lower the required Hadoop MapReduce execution time.

Based on Figure 12, when using a block size of 64 MB with 17 block chunks, the computation of Hadoop MapReduce in all the tested datasets runs the slowest compared to using block sizes of 128 MB, 256 MB and 512 MB. As previously discussed, the large number of block pieces to be processed makes namenode and MapReduce performance slower. This will hamper the performance of the cluster and the computation time will be longer. Therefore, a block size of 64 MB is not suitable for a file size of 1.06 GB. Meanwhile, the fastest computation time is shown when the block size is 512 MB with the number of blocks produced only 3 blocks.

*B. Performance Comparison of Hadoop Private Cloud Clusters and Physical Hadoop Clusters*

Generally, the performance of the two clusters depends on machine specifications, computational data size, dataset type, number of machines involved and the size of the HDFS block. To measure the significant difference between the performance of Hadoop MapReduce clusters with physical machines (without virtualization) and virtualized private clouds, t or t-test statistical tests are used. The type of t-test used is the paired sample t-test. Statistical tests will show whether the average Hadoop MapReduce computing time will experience a significant change when the cluster is virtualized with a private cloud. In this statistical test, the significance (α) is set at 5%. Then, in facilitating the calculation of the t-test, the Microsoft Excel program was used.

In scenarios varying the amount of 1.5 million data across three synthetic datasets, the private cloud cluster processes anti-correlated (55%), independent (31%) and correlated (46%) data faster than the physical Hadoop cluster (Figure 14). Furthermore, cluster performance improves significantly when the amount of data in all dataset types is increased to 2.5 million, 5 million, 8 million, 10 million, and 12 million. In this respect, a private cloud Hadoop cluster is still much faster than a physical Hadoop cluster. Overall, the physical cluster executes anti-correlated, independent and correlated data types with time ranges of 114-481 seconds, 68-156 seconds and 65-135 seconds respectively. Whereas the Hadoop private cloud cluster only takes 51-137 seconds, 47-106 seconds and 35-103 seconds respectively on the same type of data to complete the running Skyline application.

TABLE X. COMPARISON OF COMPUTING TIME IN SCENARIO 1

| Jumlah Data (Juta) | *Klaster Hadoop Private Cloud* | | *Klaster Hadoop Fisik* | | | |
|---|---|---|---|---|---|---|
| | *Correlated* | *Anticorrelated* | *Correlated* | *Anticorrelated* | *Correlated* | *Anticorrelated* |
| 1.5 | 35 | 51 | 47 | 65 | 114 | 68 |
| 2.5 | 46 | 56 | 52 | 86 | 170 | 86 |
| 5 | 50 | 63 | 53 | 87 | 142 | 90 |
| 8 | 62 | 101 | 73 | 118 | 145 | 140 |
| 10 | 70 | 129 | 78 | 132 | 349 | 146 |
| 12 | 103 | 137 | 106 | 135 | 481 | 156 |

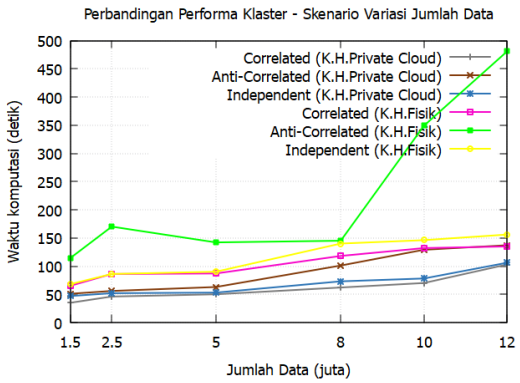


Fig. 14 Cluster Performance Comparison – Scenario 1

Based on the results of the statistical test in Figure 4.48, the resulting t count or t stat (2.93) > t table or t critical two tails (2.77) on anti-correlated data, t count or t stat (7.13) > t table or t critical two tail (2.77) on correlated data and t count or t stat (7.86) > t table or t critical two tail (2.77) on independent data. This means H0 is rejected and H1 is accepted. So, it can be concluded that the average Hadoop MapReduce processing time in response to changes in the amount of data before being virtualized (using a physical machine) is not the same as the average processing time of Hadoop MapReduce in responding to changes in the amount of data after being virtualized with a private cloud. Or in other words, in the scenario of changing the amount of data, Hadoop MapReduce shows a much better performance when virtualized with a private cloud, compared to using a physical cluster.

| | Anti-Correlated - Dengan Klaster Mesin Fisik | Anti-Correlated - Dengan Klaster Virtualisasi Private Cloud |
|---|---|---|
| Mean | 257.4 | 87.2 |
| Variance | 22994.3 | 1369.2 |
| Observations | 5 | 5 |
| Pearson Correlation | 0.84354203 | |
| Hypothesized Mean Difference | 0 | |
| df | 4 | |
| t Stat | 2.934912715 | |
| P(T<=t) one-tail | 0.021302281 | |
| t Critical one-tail | 2.131846786 | |
| P(T<=t) two-tail | 0.042604562 | |
| t Critical two-tail | 2.776445105 | |

| | Independent - Dengan Klaster Mesin Fisik | Independent - Dengan Klaster Virtualisasi Private Cloud |
|---|---|---|
| Mean | 123.6 | 72.4 |
| Variance | 1090.8 | 488.3 |
| Observations | 5 | 5 |
| Pearson Correlation | 0.905291369 | |
| Hypothesized Mean Difference | 0 | |
| df | 4 | |
| t Stat | 7.13177682 | |
| P(T<=t) one-tail | 0.00102199 | |
| t Critical one-tail | 2.131846786 | |
| P(T<=t) two-tail | 0.002043981 | |
| t Critical two-tail | 2.776445105 | |

| | Correlated - Dengan Klaster Mesin Fisik | Correlated - Dengan Klaster Virtualisasi Private Cloud |
|---|---|---|
| Mean | 111.6 | 66.2 |
| Variance | 566.3 | 514.2 |
| Observations | 5 | 5 |
| Pearson Correlation | 0.84661179 | |
| Hypothesized Mean Difference | 0 | |
| df | 4 | |
| t Stat | 7.860367148 | |
| P(T<=t) one-tail | 0.000707757 | |
| t Critical one-tail | 2.131846786 | |
| P(T<=t) two-tail | 0.001415514 | |
| t Critical two-tail | 2.776445105 | |

Fig. 15 T-test results in scenario 1

In the second scenario, namely variations in the number of machines, the private cloud virtualized Hadoop cluster has better performance when compared to the physical cluster. When executing the Skyline MR-BNL application on anti-correlated, correlated and independent data using 1 machine, the computing time with the Hadoop private cloud cluster is superior to the physical Hadoop cluster (Figure 16) by 36%, 35% and 36%, respectively. Furthermore, as the number of machines is scaled from 2 machines to 7 machines, the computing performance is constantly increasing, thus reaching an ideal state. While in the physical Hadoop cluster, the addition of a machine gives a different response to the three datasets. The anti-correlated and independent dataset achieves ideal conditions when there are 6 machines, so that when the machines are added to 7 machines, OC (Overhead Communication) occurs. OC occurs when a cluster has complexity in data distribution processes, synchronization between nodes and communication between Hadoop daemons. Slightly different from the other two datasets, the correlated dataset achieves ideal conditions when there are 5 machines. OC conditions are also found when the machines are scaled to 6 machines and 7 machines. This event shows the advantages of virtualization technology with private cloud in avoiding overhead conditions when running Hadoop MapReduce computations.

TABLE X. COMPARISON OF COMPUTING TIME IN SCENARIO 2

| Jumlah Mesin | Klaster *Hadoop Private Cloud* | | | Klaster *Hadoop Fisik* | | |
|---|---|---|---|---|---|---|
| | *Correlated* | *Anticorrelated* | *Correlated* | *Anticorrelated* | *Correlated* | *Anticorrelated* |
| 1 | 110 | 128 | 119 | 171 | 201 | 179 |
| 2 | 104 | 116 | 116 | 154 | 188 | 154 |
| 3 | 104 | 113 | 110 | 148 | 160 | 146 |
| 4 | 97 | 111 | 109 | 139 | 156 | 145 |
| 5 | 94 | 109 | 108 | 134 | 154 | 141 |
| 6 | 93 | 106 | 106 | 167 | 148 | 141 |
| 7 | 87 | 100 | 91 | 205 | 250 | 225 |

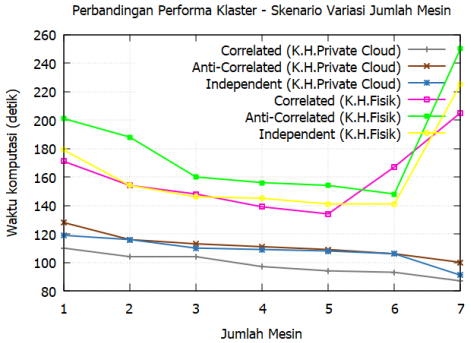


Fig. 16 Cluster Performance Comparison – Scenario 2

| | Anti-Correlated - Dengan Klaster Mesin Fisik | Anti-Correlated - Dengan Klaster Virtualisasi Private Cloud |
|---|---|---|
| Mean | 176 | 109.1666667 |
| Variance | 1508.8 | 31.76666667 |
| Observations | 6 | 6 |
| Pearson Correlation | -0.549952051 | |
| Hypothesized Mean Difference | 0 | |
| df | 5 | |
| t Stat | 3.878752663 | |
| P(T<=t) one-tail | 0.0058283 | |
| t Critical one-tail | 2.015048373 | |
| P(T<=t) two-tail | 0.0116566 | |
| t Critical two-tail | 2.570581836 | |

| | Independent - Dengan Klaster Mesin Fisik | Independent - Dengan Klaster Virtualisasi Private Cloud |
|---|---|---|
| Mean | 158.6666667 | 106.6666667 |
| Variance | 1078.666667 | 70.26666667 |
| Observations | 6 | 6 |
| Pearson Correlation | -0.848989556 | |
| Hypothesized Mean Difference | 0 | |
| df | 5 | |
| t Stat | 3.168141336 | |
| P(T<=t) one-tail | 0.012433372 | |
| t Critical one-tail | 2.015048373 | |
| P(T<=t) two-tail | 0.024866745 | |
| t Critical two-tail | 2.570581836 | |

| | Correlated - Dengan Klaster Mesin Fisik | Correlated - Dengan Klaster Virtualisasi Private Cloud |
|---|---|---|
| Mean | 157.8333333 | 96.5 |
| Variance | 668.5666667 | 44.3 |
| Observations | 6 | 6 |
| Pearson Correlation | -0.618835736 | |
| Hypothesized Mean Difference | 0 | |
| df | 5 | |
| t Stat | 4.937396094 | |
| P(T<=t) one-tail | 0.002165907 | |
| t Critical one-tail | 2.015048373 | |
| P(T<=t) two-tail | 0.004331815 | |
| t Critical two-tail | 2.570581836 | |

Fig. 17 T-test results in scenario 2

In Figure 17, because t count (9.11) > t table (2.77) on anti-correlated data, t count (8.02) > t table (2.77) on correlated data and t count (43.8) > t table (2.77) on data independent, then H1 is accepted. So, it is concluded that when running scenario 1, there is a difference between the average computing time before the cluster is virtualized and after it is virtualized with the Private Cloud. It also shows that when scaling the number of machines, Hadoop cluster virtualized with private cloud performs better using physical computers (not virtualized).

The same observation was also made for both clusters when the HDFS block size was varied (Figure 18). Along with the addition of the HDFS block size, the two clusters show the same performance trend. Both clusters show an increase in Hadoop MapReduce computation time when the number of blocks to be executed decreases. In this respect, overall, private cloud clusters complete computations faster than physical Hadoop clusters. For example, using a block size of 64 MB on all three datasets of 1.06 GB would result in a block chunk of 17 blocks. Computation time required by private cloud clusters on anti-correlated, independent and correlated data is lower than physical Hadoop clusters with respective percentages of 26%, 18% and 23%. Then, when the block size was increased to 128 MB, 256 MB and 512 MB so that the block chunks became smaller, the cluster performance decreased significantly on all types of datasets.

TABLE XI. COMPARISON OF COMPUTING TIME IN SCENARIO 1

| *Block Size* | Klaster *Hadoop Private Cloud* | | | Klaster *Hadoop Fisik* | | |
|---|---|---|---|---|---|---|
| | *Correlated* | *Anticorrelated* | *Independent* | *Correlated* | *Anticorrelated* | *Independent* |
| 64 *MB* | 103 | 118 | 113 | 133 | 160 | 137 |
| 128 *MB* | 99 | 106 | 100 | 130 | 157 | 134 |
| 256 *MB* | 91 | 105 | 93 | 128 | 132 | 129 |
| 512 *MB* | 80 | 91 | 89 | 116 | 127 | 121 |

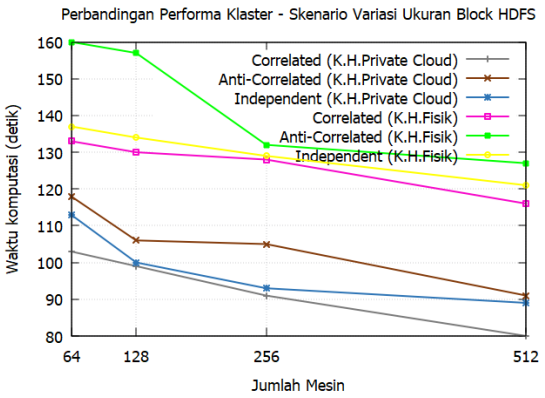


Fig. 18 Cluster Performance Comparison – Scenario 3

| | Anti-Correlated - Dengan Klaster Mesin Fisik | Anti-Correlated - Dengan Klaster Virtualisasi Private Cloud |
|---|---|---|
| Mean | 138.6666667 | 100.6666667 |
| Variance | 258.3333333 | 70.33333333 |
| Observations | 3 | 3 |
| Pearson Correlation | 0.67386737 | |
| Hypothesized Mean Difference | 0 | |
| df | 2 | |
| t Stat | 5.428571429 | |
| P(T<=t) one-tail | 0.016149255 | |
| t Critical one-tail | 2.91998558 | |
| P(T<=t) two-tail | 0.03229851 | |
| t Critical two-tail | 4.30265273 | |

| | Independent - Dengan Klaster Mesin Fisik | Independent - Dengan Klaster Virtualisasi Private Cloud |
|---|---|---|
| Mean | 128 | 94 |
| Variance | 43 | 31 |
| Observations | 3 | 3 |
| Pearson Correlation | 0.958634312 | |
| Hypothesized Mean Difference | 0 | |
| df | 2 | |
| t Stat | 29.44486373 | |
| P(T<=t) one-tail | 0.000575705 | |
| t Critical one-tail | 2.91998558 | |
| P(T<=t) two-tail | 0.001151411 | |
| t Critical two-tail | 4.30265273 | |

| | Correlated - Dengan Klaster Mesin Fisik | Correlated - Dengan Klaster Virtualisasi Private Cloud |
|---|---|---|
| Mean | 124.6666667 | 90 |
| Variance | 57.33333333 | 91 |
| Observations | 3 | 3 |
| Pearson Correlation | 0.955267051 | |
| Hypothesized Mean Difference | 0 | |
| df | 2 | |
| t Stat | 18.67895141 | |
| P(T<=t) one-tail | 0.00142693 | |
| t Critical one-tail | 2.91998558 | |
| P(T<=t) two-tail | 0.002853861 | |
| t Critical two-tail | 4.30265273 | |

Fig. 19 T-test results in scenario 3

In scenario 3, the t-test hypothesis is formulated, namely H0 = Average computational time for Hadoop MapReduce in response to changes in HDFS blocksize before virtualization (using a physical machine) = Average computational time for Hadoop MapReduce in response to changes in HDFS blocksize after being virtualized with a private cloud. While H1 = Average computational time for Hadoop MapReduce in response to changes in HDFS blocksize before virtualization (using a physical machine) ≠ Average computational time for Hadoop MapReduce in response to changes in HDFS blocksize after virtualization with a private cloud. So based on Figure 18 where t count (5.42) > t table (4.30) on anti-correlated data, t count (18.67) > t table (4.30) on correlated data and t count (29.44) > t table (4.30) on data independent, Hadoop clusters virtualized with private cloud again outperform physical Hadoop clusters on all three datasets when processing various HDFS block sizes for 1.06 GB of data

Based on the results of the t statistical test in Figure 15, Figure 17, and Figure 19, it can be proven that in carrying out the entire test starting from changing file sizes, changing the number of machines and modifying blocksize HDFS, with certain specifications, the Hadoop private cloud cluster built works better in running Hadoop MapReduce compute, versus a cluster of physical machines (no virtualization).

## V. Conclusions and Suggestion

### A. Conslusions

- To implement the Hadoop MapReduce cluster on top of private cloud computing, the process of installing and configuring the environment where the Hadoop daemon runs as well as configuration parameters for the Hadoop daemon is carried out. The Hadoop daemons in question are namenode, datanode, secondarynamenode, resourcemanager and nodemanager.
- Increasing the volume of data executed from 1.5 million to 12 million using 4 machines will cause an increase in computation time and a decrease in cluster performance.
- Increasing the number of machines from 1 machine to 7 machines increases the performance of Hadoop private cloud clusters, while for Hadoop clusters it physically causes overhead communications.
- Block Size determines the number of block pieces to be executed thereby affecting the computing speed of Hadoop MapReduce. Hadoop private cloud clusters and physical Hadoop clusters both show an increase in Hadoop MapReduce computation time when the number of blocks to be executed decreases.
- In all performance testing scenarios that the researchers conducted, the Hadoop cluster which was virtualized on a Private Cloud with the Intel(R) Xeon (R) E3-1225 v5 @ 3.30 GHz RAM 16 GB, worked much better in executing the Skyline application than Hadoop cluster built on a physical machine with the machine specs Intel Core i5 CPU @ 3.00GHz 4GB RAM. This is evidenced from the results of a comparison of the average computation time of the two clusters with the paired t-test where in the scenario of changing the amount of data it results that t count (2.93) > t table (2.77) on anti-correlated data, t count (7.13) > t table (2.77) on correlated data and t count (7.86) > t table (2.77) on independent data. In the scenario of changing the number of machines, t count (9.11) > t table (2.77) on anti-correlated data, t count (8.02) > t table (2.77) on correlated data and t count (43.8) > t table (2.77) on independent data. While the modification scenario for the HDFS blocksize size resulted in t count (5.42) > t table (4.30) on anti-correlated data, t count (18.67) > t table (4.30) on correlated data and t count (29.44) > t table (4.30) on independent data.

### B. Suggestion

Based on the results of the research that has been done, there are several suggestions that can be given, including:

- It is advisable to pay more attention to the hardware specifications of the machines in the cluster because they determine the computational performance of Hadoop MapReduce
- It is hoped that the two clusters tested will have the exact same specifications.
- We recommend that the amount of data used is larger and the types of data are more varied, not only synthetic data
- It is hoped that in the future, virtual clusters built on UNRAM's private cloud servers can be accessed from outside UNRAM
- The process of installing and configuring Hadoop and the environment is done manually on each node, due to machine specifications that make it impossible to use cluster manager tools


## Acknowledgment

First of all, I would like to express my gratitude to my supervisor who has provided direction, guidance, criticism and suggestions throughout the writing process. I really appreciate it.
I also express my deepest gratitude to the lecturers who have given valuable knowledge and experience to me during my studies. In addition, to my family and friends who have supported and motivated me to finish this thesis, thank you very much. Finally, the writer hopes that the results of this research can provide benefits to anyone who reads it.


## References


[1] A. Nuriadin, Y. Dyan Nofia Harumike, D. Tana Sanggamu, P. Studi Ilmu Komunikasi, and U. Islam Blitar, "Sejarah Perkembangan Dan Implikasi Internet Pada Media Massa Dan Kehidupan Masyarakat," *SELASAR KPI : Referensi Media Komunikasi dan Dakwah*, vol. 1, no. 1, 2021, [Online]. Available: https://ejournal.iainu-kebumen.ac.id/index.php/selasar/index

[2] N. Subagya, A. Wijajarto, and A. Almaarif, "Implementasi Dan Analisis Hadoop Element Availability Berdasarkan Daemon Log Monitoring Hadoop Element Availability Implementation And Analysis Based On Daemon Log Monitoring," *e-Proceeding of Engineering*, vol. 8, no. 5, pp. 9223–9234, 2021, Accessed: Nov. 06, 2022. [Online]. Available: https://openlibrary.telkomuniversity.ac.id/pustaka/170581/implementasi-dan-analisis-hadoop-element-availability-berdasarkan-daemon-log-monitoring-menggunakan-log4j-logging.html

[3] A. M. Ryanto, "Analisis Kinerja Framework Big Data Pada

Cluster Tervirtualisasi: Hadoop Mapreduce Dan Apache Spark," Universitas Hasanuddin, Makassar, 2017. Accessed: Oct. 09, 2022. [Online]. Available: http://digilib.unhas.ac.id/

[4] S. K. Subramanian and K. C. Gouda, "A Study on The Different Aspects Of Virtual Private Cloud," *International Journal of Applied Engineering Research*, vol. 10, no. 86, pp. 343–347, 2015, Accessed: Nov. 11, 2022. [Online]. Available: http/www.ripublication.com/ijaer.html

[5] W. S. Prabowo, M. H. Muslim, and S. B. Iryanto, "Pusat Data Privat Virtual Pemerintah Berbasis Komputasi Awan (Studi Empiris Pada Lembaga Ilmu Pengetahuan Indonesia)," *Jurnal Penelitian dan Pengembangan Komunikasi dan Informatika*, vol. 6, no. 2, pp. 1–12, 2015.

[6] O. Achahbar and M. Riduan, "The Impact of Virtualization on High Performance Computing Clustering in the Cloud," 2014. Accessed: Mar. 15, 2023. [Online]. Available: https://www.researchgate.net/publication/282531800_The_Impact_of_Virtualization_on_High_Performance_Computing_Clustering_in_the_Cloud

[7] N. Azizah and H. Saptono, "UJI PERFORMA DAN PERBANDINGAN RDBMS MYSQL DAN HIVE-HADOOP," *Jurnal Informatika Terpadu*, vol. 6, no. 1, pp. 20–28, [Online]. Available: https://journal.nurulfikri.ac.id/index.php/JIT

[8] I. P. A. P. Wibawa, I. D. Giriantari, and M. Sudarma, "Komputasi Paralel Menggunakan Model Message Passing Pada SIM RS (Sistem Informasi Manajemen Rumah Sakit)," *Majalah Ilmiah Teknologi Elektro*, vol. 17, no. 3, p. 439, Dec. 2018, doi: 10.24843/mite.2018.v17i03.p20.

[9] V. Starostenkov and K. Grigorchuk, *Hadoop Distributions: Evaluating Cloudera, Hortonworks, and MapR in Micro-benchmarks and Real-world Applications*. 2013. [Online]. Available: www.altoros.com

[10] M. A. Siddique, H. Tian, M. Qaosar, and Y. Morimoto, "MapReduce algorithm for variants of skyline queries: Skyband and dominating queries," *Algorithms*, vol. 12, no. 8, 2019, doi: 10.3390/a12080166.

[11] P. M. Mell and T. Grance, "The NIST definition of cloud computing," *NIST Computer Security Resource Center CSRC*, 2011, doi: 10.6028/NIST.SP.800-145.

[12] T. White, *THIRD EDITION Hadoop: The Definitive Guide*, 3rd ed. United States of America.: O'REILLY, 2022.

[13] F. I. K. U. Mahasiswa Peserta Mata Kuliah Komputasi Parallel Lanjut, *Kajian Tematik Infrastruktur Cloud Computing*. Depok, Indonesia: FAKULTAS ILMU KOMPUTER UNIVERSITAS INDONESIA, 2018. [Online]. Available: www.cs.ui.ac.id

[14] "Hadoop Cluster," *Databricks*, 2022. https://www.databricks.com/glossary/hadoop-cluster (accessed Sep. 09, 2022).

[15] A. L. Ramdani, "Pemilihan Akun Berpengaruh Pada Data Twitter Menggunakan Skyline Query Dalam Mapreduce Framework," Sekolah Pascasarjana Institut Pertanian Bogor, Bogor, 2016. Accessed: Oct. 09, 2022. [Online]. Available: https://repository.ipb.ac.id/

[16] 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, Jul. 2022, doi: 10.1109/TKDE.2020.3021470.

[17] A. Zaman and Y. Morimoto, "MapReduce-Based Computation of Area Skyline Query for Selecting Good Locations in a Map," in *2017 IEEE International Conference on Big Data (Big Data)*, Boston, MA, USA: IEEE, Jan. 2017, pp. 3–5. doi: 10.1109/BigData.2017.8258540.

[18] B. Zhang, S. Zhou, and J. Guan, "Adapting Skyline Computation to the MapReduce Framework: Algorithms and Experiments," in *International Conference on Database Systems for Advanced Applications*, 2011, pp. 403–414. doi: https://doi.org/10.1007/978-3-642-20244-5_39.

[19] I. Made, S. W. Putra, H. Wijayanto, and A. Zafrullah, "MENGGUNAKAN SKYLINE QUERY PADA LOKASI WISATA DI PULAU LOMBOK (Parallel Computing For Calculation Of Domination Relations Using Skyline Query On Tourism Locations in Lombok Island)."

[20] T. Ivanov, R. V, Zicari, I. Izberovic, and K. Tolle, "Performance Evaluation of Virtualized Hadoop Clusters," Frankfurt am Main, Germany, Nov. 2014.