CHAPTER ONE

1.1 Background of the study

Big data is data with high velocity, high volume, and high variety that can be used to discover hidden patterns when analysed. 80% of this large data is unstructured while the remaining 20% is structured. (Rui. 2014)

Benchmark means making meaningful comparisons to others and identifying opportunities to improve.

Big data application benchmarking means using useful data to improve the performance of the application and comparing the application with other applications and decide which application is the leader in the field. Business owners use big data application benchmarks to identify the leading business in the field and use the benchmark data to close the gap between them. (Rui, H. 2014)

Benchmarks for big data can be defined by the 3v. (Rui. 2014)

- High volume: can the benchmark test the scalability of the system to huge volumes of data
- High velocity: can the benchmark test the system's ability to deal with real-time data at a high velocity
- High variety: can the benchmark test the system ability to include structured data, unstructured data, and semi-structured data

Big data benchmarks are developed to evaluate and compare the performance of big data systems and architectures.

The big data benchmark process; (Rui. 2014)

- Planning: Determine the evaluation metrics
- Generation of data: The data in the evaluation is generated
- Generation of test: The test in the evaluation is generated
- Execution: The evaluation of the benchmark test is reported.
- Analysis and evaluation: The result of the benchmark is analysed and evaluated.

Big Data benchmark techniques;

Data Generation techniques: Data generation techniques are reviewed according to the 3V(volume, velocity, variety) characteristics of big data. Comparison of data generation techniques in existing big data benchmarks. (Rui, H. 2014)

Benchmark efforts	Volume	Velocity	variety
HiBench	Partially scalable	Uncontrollable	Text
GridMix	Scalable	Uncontrollable	Texts
PigMix	Scalable	Uncontrollable	Texts
YCSB	Scalable	Uncontrollable	Texts
TPC-DS	Scalable	Semi controllable	Tables
BigBench	Scalable	Semi controllable	Texts, web logs, tables
LinkBench	Partially scalable	Semi controllable	Graphs
CloudSuite	Scalable	Semi controllable	Texts, graphs, video, tables
BigDataBench	Scalable	Semi controllable	Texts resumes, graphs, tables

Benchmarking techniques (Rui, H. 2014)

Benchmark efforts	Workloads		Software stacks
	Type	Example	
HiBench	Office analytics	Sort, WordCount, TeraSort, PageRank, K-means, Bayes classification	Hadoop and Hive
	Realtime analytics	Nutch indexing	
GridMix	Online services	Sort, sampling a large dataset	Hadoop
PigMix	Online services	12 data queries	Hadoop
YCSB	Online services	OLTP (read, write, scan, update)	NoSQL systems
BigDataBench	Online services	Database operations (read, write, scan)	NoSQL system DBMS, real-time

	Offline analytics Real-time analytics	Micro Benchmarks (sort, grep, WordCount, CFS); search engine (index, PageRank); Social network (K-means, connected components (CC). Relational database query (select, aggregate, join)	and offline analytics systems

1.2 Problem of the statement

Real-world data that are meant to be the input for workloads are not available because data owners are not willing to share that data due to confidential issues or data protection laws. (Khushboo .2017)

Lack of skilled workers executing the benchmarks is a major factor that has given rise to Inaccurate data used in benchmarks.

Company's Run benchmark that test workloads different from what customers expect to see in production. The company's also Running benchmarks on hardware significantly different than what your customers are expected to use.

Big data is generating a lot of data rapidly every year and traditional storing tools can handle this data.

The high velocity of information coming in when we don't have the right technology to handle it. (Khushboo, W.2017)

Data provided are not in context with the needed data for benchmarking. (Khushboo, W.2017)

1.3 Aim and objective.

The aims and objectives of the study are ;

a. Explain what big data benchmark means.
b. Compare data generation techniques in big data and benchmarking techniques of different benchmarking suites.

c. Perform big data benchmark between two cloud platforms.
d. Provide a representation of the real-world application scenarios as closely as possible, provide repeatability and comparability of results, and would be easy to execute.

1.4 Justification of the study

The reason why this study was conducted is to show how the importance of benchmarks in big data.

1.5 Significance of the study

The study has shown me the importance of big data benchmark in an application that assists system owners to make decisions for planning system features, tuning system configurations, validating deployment strategies, and conducting offer efforts to improve the system.

The study helps in making the product features correspond to users implied or stated needs and impacting their satisfaction

The study shows Big data benchmarks are of industrial significance because they apply to the actual and emerging needs of specific industries and specific company-size segments

1.6 Methodology overview

- Reliable tool like Hadoop for my benchmark.
- Cloud platform used is secure.
- Private cloud using a hybrid model was expanded.
- Experienced data analyst was hired.

1.7 Organization of subsequent chapters

- Chapter One, Introduction: This is the current chapter and it introduces the project.
- Chapter Two, Literature Review: In this chapter, we shall discuss and review all the necessary theories and applications of big data benchmarking.
- Chapter Three, Methodology: In this chapter, we will discuss all used methods of solving big data benchmarking issues.
- Chapter Four, Report: in this chapter, we report the benchmark of Microsoft Azure and Amazon EC2 cloud platforms.

- Chapter Five, Summary and Conclusions: This chapter shall conclude the project, summarizing, software testing and giving a recommendation.

Chapter Two: Literature review

It is very significant to review different but relevant perspectives and literature from various academic works as they relate to the conceptual framework of this research work and also by articulating the key concepts of the research work.

The literature surrounding the Big data application benchmark outlines the strategy that should be used and points out some truths and trends. Scholars have collectively defined a single truth throughout all of the literature regarding what personality should be utilized. They all harbour the same belief regarding personnel use for big data application benchmark.

Although the scholars believe in similar personnel being deployed, they differ on how the big data application benchmark should be used, which influences and illustrates differences in strategy and tactics.

Studies using Big Data Benchmarks Rui Han and Xiaoyi Lu. in 'On Big Data Benchmarking' discuss the vital requirements, challenges, and tests in developing big data benchmarks and their execution .These are relevant when considering the 4V (Volume, Velocity, Variety, and Veracity) properties, generating workloads, and test execution in big data systems. Methodologies like Layer design, Data generation, and Test generation are designed to address these requirements challenges. This paper compared data generation techniques in existing big data benchmarks According to the authors, workloads in current big data benchmarks are in three categories Online services, Offline services, Real-time services. Big data systems have been developed to manage and process big data efficiently, and these have given growth to various new requirements for developing a new group of big data benchmarks. (Rui and Xiaoyi ,2014)

Chapter Three: Methodology.

We can handle the massive volume of data generated from big data by using tools like Hadoop that can manage structured, semi-structured and unstructured data. Purchasing a robust hardware component enables an increase in memory and powerful parallel processing to process high volumes of data swiftly. (Khushboo ,2017)

We can ensure data providers their data is secure by examining the security of our cloud providers. We should make sure that our cloud platform providers have frequent security audits and have a disclaimer that includes paying penalties in case adequate security standard is not met. We should create a policy that allows only authorized users access to the data. (Khushboo, W.2017)

The high velocity of data coming in can be controlled by expanding private cloud using a hybrid model allows arising the need for additional computational power needed for data analysis and to select hardware, software, and business process changes to handle high-pace data need. (Khushboo, W.2017)

Big data benchmarks that are performed should display the hardware configuration of the system used so that business owners will not misinterpret the result of the benchmark.

Human resource managers must ensure that the candidates that they are hiring are well experienced and have written a professional examination recently, this needs to be done to prevent hiring an inexperienced worker.

Chapter Four: Report.

Big data benchmark for Amazon EC2 and Microsoft Azure cloud platforms using the HIBench benchmark suite, which includes the workload examples: MicroBenchmarks (Sort, WordCount), SQL Benchmarks(Aggregation, join,), Web Search Benchmark (Page Rank), and Machine Learning Benchmarks (Bayes and K-Means). The response time per benchmark value is in seconds and the throughput value is in megabytes per sec, as measured by incrementing the number of nodes by one from one to five. By changing the dataset size (1GB, 100GB, and 1,000GB) to represent big application computation using Hadoop, and by using each benchmark (Sort, WordCount), SQL Benchmarks (Aggregation, Join), Web Search Benchmark (Page Rank), and Machine Learning Benchmarks (Bayes, K-Means), the table shows the performance of Amazon EC2 and Microsoft Azure cloud platforms. The test was executed once. (Karthika ,2017)

Table 1a: WordCount Response time(s) between Azure and EC2 (Karthika, 2017)

Data Size	1GB		100GB		1,000 GB	
Nodes	Azure	EC2	Azure	EC2	Azure	EC2
1	38.494S	49.81S	109.529S	75.963S	1,074.223S	968.444 S
2	68.605S	73.857S	231.708S	294.892S	3,384.274S	3,016.376S
3	42.336S	47.862S	101.938S	197.763S	1,622.662S	1,724.454S
4	36.544S	37.57 S	61.689 S	77.794 S	809.162 S	805.737 S
5	39.036S	46.678S	80.793 S	90.083 S	1,194.508S	1,225.565S

- Table 1b: WordCount throughput performance (mb/s) between Azure and EC2 (Karthika, 2017)

Data Size	1GB		100GB		1,000 GB	
Nodes	Azure	EC2	Azure	EC2	Azure	EC2
1	14,963,013	13,898,957	19,027,962	14,951,001	2,353,937	2,458,881
2	24,247,449	21,447,752	43,250,980	22,294,045	4,909,444	4,301,018
3	26,296,800	21,991,727	54,570,933	48,942,960	6,669,215	6,051,829
4	20,609,073	26,667,440	40,253,705	58,040,250	7,415,952	7,656,992
5	28,090,250	27,323,432	71,470,783	56,674,772	9,845,391	9,205,298

Azure performed better than EC2 cloud platform small data size. When the data size is increased, they have similar performance.

Table 2a: Sort response time(s) between Azure and EC2 (Karthika, 2017)

Data Size	1GB		100GB		1,000 GB	
Nodes	Azure	EC2	Azure	EC2	Azure	EC2
1	31.481	36.687	117.8	146.569	1,785.362	1,665.656
2	29.573	28.856	67.939S	103.115	921.965	956.111
3	23.478	29.122	39.498	84.009	670.769	719.021
4	22.756	26.908	36.532	81.968	556.199	565.557
5	20.453	25.644	36.915	55.139	442.98	473.28

Table 2b: Sort throughput performance(mb/s) between Azure and EC2 (Karthika, 2017)

Data Size	1GB		100GB		1,000 GB	
Nodes	Azure	EC2	Azure	EC2	Azure	EC2
1	32,608,106	27,980,892	37,427,237	30,081,035	4,154,276	4,452,885

Azure cloud platform performed better than EC2 cloud platform for data size 1GB, and 100GB. When increased to 1,000Gb they have similar performance.

EC2 (Karthika, 2017)

Data Size	Uservisits:1,000,000 Pages: 120,000		Uservisits:10,000,000 Pages: 1,200,000		Uservisits:100,000,000 Pages: 12,000,000	
Nodes	Azure	EC2	Azure	EC2	Azure	EC2
1	38.564	47.471	64.428	104.162	449.942	402.174
2	32.888	33.335	54.298	50.135	316.869	310.544
3	34.596	32.445	40.659	45.913	186.598	225.432
4	31.85	30.868	39.147	42.538	161.46	172.801
5	31.006	30.647	41.523	41.861	121.12	171.525

Azure and EC2 (Karthika, 2017)

Data Size	Uservisits:1,000,000 Pages: 120,000		Uservisits:10,000,000 Pages: 1,200,000		Uservisits:100,000,0 Pages: 12,000,000	
Nodes	Azure	EC2	Azure	EC2	Azure	EC2
1	966,619	781,957	5,779,898	3,575,078	8,193,185	9,166,
2	1,133,444	1,118,245	6,858,213	7,427,691	11,634,013	11,870
3	1,077,486	1,148,920	9,158,791	8,110,715	19,756,148	16,352
4	1,170,383	1,207,616	9,512,537	8,754,226	22,832,019	21,333
5	1,202,241	1,216,324	8,968,217	8,895,804	30,436,408	21,492

Azure and EC2 cloud platform have the same performance for all pages.

Table 4a: join response time (s) between Azure and EC2 (Karthika, 2017)

Data Size	Uservisits:1,000,000 Pages: 120,000		Uservisits:10,000,000 Pages: 1,200,000		Uservisits:100,000,000 Pages: 12,000,000	
Nodes	Azure	EC2	Azure	EC2	Azure	EC2
1	78.91	88.309	64.428	104.162	449.942	402.174
2	56.888	62.316	54.298	50.135	316.869	310.544
3	54.894	63.223	40.659	45.913	186.598	225.432
4	53.526	61.405	39.147	42.538	161.46	172.801
5	53.945	60.443	41.523	41.861	121.12	171.525

Table 4b: join throughput performance (mb/s) between Azure and EC2 (Karthika, 2017)

Data Size	Uservisits:1,000,000 Pages: 120,000		Uservisits:10,000,000 Pages: 1,200,000		Uservisits:100,000,000 Pages: 12,000,000	
Nodes	Azure	EC2	Azure	EC2	Azure	EC2
1	12,678	11,328	70,522	61,605	191,463	222,571
2	17,586	16,054	118,502	102,871	321,544	411,641
3	18,224	15,823	133,252	120,122	516,412	439,173
4	18,690	16,292	136,028	124,868	533,100	523,644
5	18,545	16,551	143,393	125,380	669,849	620,634

Azure cloud platform performs better than EC2 cloud platform for the dataset (user visits: 1,000,000; pages: 120,000), and (user visits: 10,000,000; pages: 1,200,000). For big datasets (user visits: 100,000,000; pages: 12,000,000), both Azure and EC2 show no noticeable difference.

Data Size	Pages: 100,000		Pages: 500,000		Pages: 1,000,000	
Nodes	Azure	EC2	Azure	EC2	Azure	EC2
1	72.142	81.123	99.811	108.332	154.619	149.403
2	48.072	57.442	68.356	75.27	96.984	97.122
3	45.32	51.213	62.558	67.311	82.171	84.755
4	43.464	49.021	59.826	60.354	68.16	74.055
5	44.005	50.016	56.098	63.882	57.611	70.464

Data Size	Pages: 100,000		Pages: 500,000		Pages: 1,000,000	
Nodes	Azure	EC2	Azure	EC2	Azure	EC2
1	5,207,868	4,631,313	18,851,389	17,368,607	24,334,993	25,184,583
2	7,815,485	6,540,615	27,526,128	24,997,688	38,796,629	38,741,504
3	8,290,071	7,336,145	30,077,304	27,953,470	45,790,514	44,394,458

Azure performed better than EC2 cloud platform for larger dataset (pages: 1,000,000), and (pages: 10,000,000) in terms of both response time and throughput metric values. However, for a smaller dataset (pages: 500,000) both clouds performed

Table 6a: k-Means response time (s) between Azure and EC2 Karthika, M.,(2017)

Data Size	No. of Samples: 20,000,000 Samples\Input files:4,000,000		No. of Samples: 80,000,000 Samples\Input files: 6,000,000		No. of Samples: 100,000,000 Samples\Input files: 8,000,000	
Nodes	Azure	EC2	Azure	EC2	Azure	EC2
1	218.452	221.719	477.846	572.877	1,153.92	2,269.021
2	123.575	122.049	376.277	544.284	552.377	606.333
3	93.396	101.169	353.736	328.871	382.072	542.956
4	69.448	82.811	206.232	314.64	262.047	373.57
5	65.934	87.722	307.307	252.769	291.266	354.825

Table 6b: K-means throughput performance (mb/s) between Azure and EC2 Karthika, M.,(2017)

Data Size	No. of Samples: 20,000,000 Samples\Input files: 4,000,000		No. of Samples: 80,000,000 Samples\Input files: 6,000,000		No. of Samples: 100,000,000 Samples\Input files: 8,000,000	
Nodes	Azure	EC2	Azure	EC2	Azure	EC2
1	18,385,602	18,114,693	37,823,176	31,548,945	10,620,523	20,883,756
2	32,501,490	32,907,862	48,032,842	33,206,267	39,744,144	43,626,362

Azure performed better than EC2 cloud platform for dataset (samples: 20,000,000) in terms of response time, and for dataset (samples: 100,000,000) in terms of throughput. For larger dataset (samples: 20,000,000) in terms of throughput, (samples: 80,000,000) in terms of both response time and throughput, and (samples: 100,000,000) in terms of response

Table 7a: Bayes response time (s) between Azure and EC2 Karthika, M.,(2017)

Data Size	Pages: 100,000		Pages: 500,000		Pages: 1,000,000	
Nodes	Azure	EC2	Azure	EC2	Azure	EC2
1	72.142	81.123	99.811	108.332	154.619	149.403
2	48.072	57.442	68.356	75.27	96.984	97.122
3	45.32	51.213	62.558	67.311	82.171	84.755
4	43.464	49.021	59.826	60.354	68.16	74.055
5	44.005	50.016	56.098	63.882	57.611	70.464

Table 7b:Tabulated Bayes response time (mb/s) between Azure and EC2 Karthika, M.,(2017)

Data Size	Pages: 100,000		Pages: 500,000		Pages: 1,000,000	
Nodes	Azure	EC2	Azure	EC2	Azure	EC2
1	5,207,868	4,631,313	18,851,389	17,368,607	24,334,993	25,184,583
2	7,815,485	6,540,615	27,526,128	24,997,688	38,796,629	38,741,504
3	8,290,071	7,336,145	30,077,304	27,953,470	45,790,514	44,394,458
4	8,644,074	7,664,185	31,450,807	31,175,664	55,203,232	50,808,890
5	8,537,803	7,511,716	33,540,875	29,453,931	65,311,352	53,398,222

Azure performed better than EC2 cloud platform as Azure cloud shows better performance metrics than EC2 for the dataset (pages: 100,000) and (pages: 500,000). For the larger dataset of (pages: 1,000,000) both the Amazon EC2 and Azure clouds performed about the same.

The results from the benchmark show that Microsoft Azure was appropriate for a smaller dataset of big database application computation up to 100gb. Testing results with Aggregation, and K-means, Bayes benchmarks revealed that both the Microsoft Azure and Amazon C2 cloud platforms performed about the same. PageRank showed that the Microsoft Azure cloud showed better performance than Amazon EC2, with better response time and throughput values compared to EC2 Microsoft Azure and Amazon EC2 showed that one cloud is equally competitive in performance to another cloud platform and that both performed about the same concerning big data application computations.

The test was conducted with different hardware configuration (Karthika,2017)

Hardware Configuration		
	Microsoft Azure	Amazon EC2
Instance type	G3	I2.2xlarge
Processor	Intel Xeon E5 v3	Intel Xeon E5-2670 v2 2.5 GHz
Memory	112GB	61GB
Storage Drives	1.5TB	1.6TB (2 *800 GB SSD)
I/O Performance	Very High/500 Mbps	High /1 Gbps

Chapter five: Summary ,Conclusion ,Recommendation.

In summary big data benchmarks is very important because it can be used to compare the performance of the application with other applications and decide which application is the leader in the field. Having this type of knowledge is very important because owners can see note where their company is lagging and improve where improvement is needed. The benchmark data used should 3V characteristics of big data. The process of performing big data benchmarks starts by planning, generating the data ,generating the test , evaluation , and analysis.

Conclusion

Every businessowner want to have an edge over their competitors

1.1 Recommendation

Big data benchmark is important because of competition

Competition is a fact of life.

Benchmarks measure performance and achievements

Big data benchmarks are developed to evaluate and compare the performance of big data systems and architectures.

Make sure that you are making a fair comparison

Benchmark is best scored on a scale of either 1-5 or 0-100%

Big data benchmarks measure performance energy efficiency and cost-effectiveness

There are three main streams of application for big data, search engines, social network, and e-commerce.

Reference

Karthika, M.,(2017).Performance Evaluation of Hadoop based Big Data Applications with HiBench Benchmarking tool on IaaS Cloud Platforms. *UNF Graduate Theses and Dissertations. 771.* p34-35;38;44;47;50;53;56;59;61. Accessed 18 October, 2020. https://digitalcommons.unf.edu/cgi/viewcontent.cgi?article=1817&context=etd

Khushboo, W.,(2017).Big Data challenges and solutions. Accessed 18 October, 2020. https://www.researchgate.net/publication/313819009_Big_Data_Challenges_and_Solutions#pf6

Rui, H. and Xiaoyi, L.,(2014).On Big Data Benchmarking. Accessed 18 October, 2020. http://barbie.uta.edu/~hdfeng/bigdata/Papers/On%20Bigdata%20Benchmarking.pdf