2020 International Conference on Data Analytics for Business and Industry: Way Towards a Sustainable Economy (ICDABI) | 978-1-7281-9675-6/20/$31.00 ©2020 IEEE | DOI: 10.1109/ICDABI51230.2020.9325662

A Survey On Cloud-Based Distributed Computing System Frameworks

Ahmed Alalawi
College of Information Technology
University of Bahrain
Sakhir,Bahrain
alalawi87@gmail.com

Alauddin Al-Omary
College of Information Technology
University of Bahrain
Sakhir,Bahrain
aalomary@uob.edu.bh

***Abstract*—The use of data is increasing steadily in the modern era of technology. That produced the term big data. Many companies are interested in analyzing this data, which amounts to several terabytes. The term distributed computing system appears as an effective technique for analyzing big data. Businesses have sought to use cloud resources to implement distributed computing in order to reduce costs and increase productivity. Distributed computing system faces the hurdle of significant material cost for providing related equipment. To avoid the financial cost problem, researchers have developed frameworks and tools for implementing distributed computing operations by using the redundant resources of cloud computing. This paper presents the advantages and disadvantages of a distributed computing system as well as analyzes traditional systems of distributed computing and concludes with a survey of frameworks for using cloud computing resources to implement distributed computing system operations at a lower cost.**

Keywords— Distributed Computing System, cloud, Hadoop, Spark, MapReduce

I. INTRODUCTION

The distributed computing system solves computational problems and processes large and redundant data by connecting a network of different computing devices to perform tasks. Hardware connected to a distributed computing network is considered as a single system with enormous capabilities that enable them to process big data and parallel tasks in a fast and distinct manner. The performance of a distributed computing system improves as the number of devices associated with it increases. The distributed computing system can perform a task in one hour compared to a single computer that takes hundreds of hours to perform the same task.

Distributed Computing System (DCS) according to [1] is a system with multiple components located on different machines that communicate and coordinate actions in order to appear as a single coherent system to the end-user. Due to modern technologies, it has become possible to connect several different computing devices in one system. All computer devices can be connected via a wired or wireless network. Computers can also be linked in a single or multiple geographical area with the ability to add and remove devices from the system.

Distributed computing frameworks are varied. The Hadoop and Spark systems are the most popular distributed computer management systems. A distributed computer needs a large infrastructure of high-specification devices to perform the analyzes of big data accurately and quickly. To solve needs a huge budget to implement Distributed computing system, cloud computing can be used as infrastructure. Hence, researchers used the excess resources in cloud computing as infrastructure to run the distributed computing system. The next section talks about the famous distributed system frameworks. Advantages of distributed computing in the third section. The fourth section talks about disadvantages of distributed computing, while the fifth section analyzes the frameworks that researchers have introduced for using cloud resources to build distributed computing.

II. DISTRIBUTED SYSTEM FRAMEWORKS

Processing Big Bata and solving computational problems is a difficult task because it is constantly increasing daily. Researchers have been interested in developing frameworks of a distributed computing system for analyzing and manipulating data. Many frameworks are designed to manage a distributed computing system efficiently and effectively.

Apache Hadoop [2] is a widespread framework to solve problems and processing Big Data using a distributed computing system. Hadoop is an open-source program that contains many tools. Hadoop is designed to be able to handle and fix common faults in a distributed computing system [3]. Dean and Ghemawat [4] explain that the Apache Hadoop model is divided into two layers. The first layer is concerned with controlling storage methods and is called Hadoop Distributed File System (HDFS). The second layer is to process and analyze stored data via MapReduce model. They describe that the Hadoop program splits data files into blocks then share the blocks of data between interconnected nodes in the distributed computing system. After that, Hadoop processes data in parallel. Nodes are the connected devices in the distributed computing system. Nodes are associated with each other locally in a group called a cluster. Otherwise, nodes that are shared across different networks geographically called a grid.

Hadoop nodes are divided into master and slave. Work in [5] explains the node's tasks that are divided into Job Tracker, Task Tracker, NameNode, and DataNode. NameNode task is to arrange files, define the system tree, and specify the naming of the slave nodes. Job Tracker task is to manage and distributes jobs on the slave nodes. The JobTracker manages resources and makes sure that tasks are executed and restart slave nodes in case the error occurs. TaskTracker is to execute the task on that node and defines the outcome of the operations. The DataNode task is to store and retrieve data in that node. The Master node usually has a the JobTracker and the NameNode task. Slave node known as Worker nodes has either DataNode task to deal with data or TaskTracker task to

978-1-7281-9675-6/20/$31.00 ©2020 IEEE
Authorized licensed use limited to: BOURNEMOUTH UNIVERSITY. Downloaded on June 27,2021 at 02:07:12 UTC from IEEE Xplore. Restrictions apply.

process data or both. The system contains an assistant master to ensure high availability. Initially, files are distributed across connected nodes into the distributed computing system. Then a table containing the locations of that data and files is created using HDFS [6]. HDFS repeats data over several nodes. Data redundancy in different nodes reduces the vulnerability of the distributed computing system which is always affected by nodes termination.

MapReduce is a processing layer in Hadoop to run parallel operations across connected nodes in the distributed computing system. MapReduce manages the methods of data transfer and results between parts of the distributed computing system. Moreover, MapReduce provides redundancy to overcome any failure and fault tolerance [7]. The work in [8] shows that MapReduce has several important characteristics such as its ability to tolerate errors and its scalability to meet business requirements. Besides, MapReduce is flexible and simple to program. MapReduce can sort petabytes Big Data in just hours. MapReduce's work consists of three steps: Map, Shuffle and Reduce. The first step, Map is the process of mapping data by sorting local data in each node to reduce network traffic. The second step, Shuffle is a process of the master node to redistributes data between nodes. The final step, reduce is to do operations with merging duplicates, reducing tasks, and filtering results to generate summaries.

There are some shortcomings when using Hadoop. The Works of [9] and [10] presents limitations in the MapReduce API that negatively affected system management. In addition, centralization in management is affected significantly due to increased scheduling volume. Additionally, MapReduce's performance in simultaneous execution and real-time processing is not efficient enough. Hence, the work of [11] shows that Hadoop YARN is a suitable alternative to MapReduce. This allows for more comprehensive processing, such as batch processing, stream processing, and charting. YARN provides group resource management and application scheduling. Works of [12] , [13], [14] and [15] pointed out that the large company have developed the Hadoop system according to work requirements and to deal with the Big Data.

Apache Spark [16] is another open-source framework of the distributed computing system. Spark is built on Resilient Distributed Dataset (RDD) with Dataset API as an interface. Spark was developed to solve problems related to Hadoop. Spark can relate to a wide variety storage system, including HDFS, the MapR File System (MapR -FS), Amazon S3, Apache Kudu, and the custom file system. The Spark is not an upgraded version of the Hadoop, but it relies on the HDFS to store data. Spark relies on the node's RAM to speed up application processing. Spark deployment is divided in three ways. Standalone deployment Spark is to process data using HDFS storage. The second, Spark is running on top of Hadoop Yarn without the need to pre-install for Spark. Finally, Spark in MapReduce (SIMR) in which Spark run in a MapReduce environment.

Researchers compared the Spark and Hadoop systems for performance and speed. Salloum et al. [17] present the problem with Hadoop HDFS is that it relies on reading and formatting data from the hard disk and then writing it back in that disk. This technique making data sharing slower. Thus, RDD in Spark read, write, and share data in memory instead of the disk. Memory is faster to share data than the hard disk. Mavridis & Karatza in [18] compares Spark and Hadoop. The authors stated that Spark is faster and more efficient due to its use of memory rather than a hard disk. The author also explained that Spark saves energy due to its ability to process data with less time than Hadoop. The works of [19], [20], [21] and [22] indicated that the Hadoop and MapReduce models do not support complex algorithms and graphs efficiently in real-time processing. Different queries take several stages. All stages rely on sharing data across the hard drive to share it across the distributed computing system. Hard disk read and write time is up to 90% of query execution time. At the other end, Spark is using RAM to share data 40 times faster than Hadoop. Therefore, works in [23] and [24] indicated that Spark works efficiently if the necessary RAM is available to store data and this may increase the material cost compared to relying on the hard disk. If the large memory is not provided, the Spark system will not perform its work properly. Hedjazi et al. [25] applied it to image processing and showed that Hadoop is better in the absence of repetitive data, while Spark is better in the presence of repetitive data such as the word count.

Based on the above, Hadoop is considered less expensive than the Spark framework, but at the same time, it is slower to operate. Framework selection depends on the company's needs and the financial cost allocated to it.

III. ADVANTAGES OF DISTRIBUTED COMPUTING

In designing a distributed computing system, large companies are obligated to focus on the duration of the system's work. The system is designed to operate with up to 99% of efficiency. The design of the distributed computing system focuses on reliability and system redundancy. Besides, the distributed computing system must be available. Hence, various resources can be easily accessed at any time. Another feature of reliability is that the system can handle multiple errors in operations [26] [27].

The distributed computer system can add and exclude devices at any moment. Therefore, work in [28], [29], and [30] explain that the operating system of the distributed computer must be able to adapt to the changes required to improve performance. The operating system should be more flexible to handle the different operating systems for each device connected to the distributed computer system. Work in [31]and [32] focus that the process of adding and replacing resources is easy and without losing any data or change in the ability of process information. In addition, these changes occur without the end-user feeling them. The criterion for measuring system tolerance to change is important for scalability. This criterion determines how efficient the system is in keeping pace with new resources. Moreover, they explain that the distributed computing system can expand and reduce the administrative powers of the system.

Optimal use of all available resources is one of the features of the distributed computing system. Applications can be executed on an idle connected node instead of using a busy node connected to the distributed computing system. Storage devices can also be combined to become a unified storage device for the network with huge capacity [33]. Besides, the distribution of the task among several devices in the network significantly improves performance [34].

Authorized licensed use limited to: BOURNEMOUTH UNIVERSITY. Downloaded on June 27,2021 at 02:07:12 UTC from IEEE Xplore. Restrictions apply.

IV. DISADVANTAGES OF DISTRIBUTED COMPUTING

There are many shortcomings in the distributed computing system. The speed of transmitted information over the network is one issue as van Steen and Tanenbaum explained [1]. The distributed computer system depends on the internal or external network for transfer data and passes a message between nodes. In the event of failure or weakening of the network, it may lead to stopping the whole process of the distributed computing system. Therefore, companies seek to provide multiple and alternative networks to obtain the best communication between distributed nodes. Especially for nodes distributed in a different geographical system and rely on the Internet.

The most applications of the distributed computing system are based on the internet. Thus, work in [35] demonstrates some causes of lack in system reliability, such as losing, delaying, or damaging messages between nodes. Likewise, the likelihood of external attacks and attempting to control by infiltrators will be very high, such as "Byzantine failure".

The distributed computer replicates data in several connected nodes for keeping the data redundant. Edinger [36] describe that the problem occurs when one of the copies is changed or updated, these modifications should be moved over all the other versions to be equal and consistent. This process of repeating operations on copies takes some time. This update may take a few seconds, which prevents any other process from running on the system until all copies have been updated. Hence, the replicating data cannot be hidden from the end-user, but this problem can be reduced by using high network speed to transfer updates faster.

V. CLOUD-BASED DISTRIBUTED COMPUTING SYSTEM FRAMEWORKS

The main goal of companies during the use of cloud computing is to reduce cost and speed up the process of distributed data processing. Hence, researchers developed several models to reach this goal. In this subsection, many models developed in this field will be surveyed and investigated.

A. Frameworks to operating distributed computing systems on cloud computing

Researchers have developed several models based on the Spark framework to deal with cloud computing. Sharma et al in [37] developed a business framework called Flint based on Spark technology. This model develops both batch handling and Streaming data. Flint improves efficiency by choosing the right price for resources and moving to other resources. The results of applying this Framework may have reduced the cost up to 90% compared to the price on demand. Additionally, running time increased by 2%. Exosphere [38] is designed on the Flint model with improvement. When Flint needs to 4000-line code the exosphere needs only 300 lines to code. Exosphere uses Server portfolios that choose the best mix of servers and the cost and save them in the portfolio. When revocation, the exosphere chose a new server to lease from the portfolio to complete work. Exosphere provides over 80% of the price when compared to the price on demand. Exosphere can deal with different parallel frameworks such as Hadoop, Spark, MPI, and BOINC. Exosphere depends on the portfolio construction technique that gives the suggestion of leasing based on the historical price trend. However, this technique may give the wrong suggestion with the frequently changing price in real-time. Xu et al. in [39] provide iSpot framework based on Spark. iSpot predicts price using Long Short-Term Memory (LSTM). iSpot decrease cost by up to 83.8% in comparison to other strategies. Liu& XU [40]offers a system data analysis tool called Elasecutor. Elasecutor Scheduling job reduces the formation range to 42%, the average application completion time to 40%, and increases batch usage by 55%. In contrast, the SplitServe-Spark [41] relies on developing the Spark system by using Serverless products such as AWS Lambdas rather than VM instances. SplitServe-Spark develops performance by 31-55% for workloads. SplitServe-Spark reduces the cost by 21% with 40% improvement in execution time. Similarly, NetSpark [42] is an improved version of Spark, reduces task execution time. NetSpark combines the management of the network buffer and RDMA (hardware-supported Remote Direct Memory Access) technology on the network card. NetSpark improves operations at Spark by 40% using a legacy network stack. . From the above, it is found that many researchers have developed various frameworks to reduce the cost of using Spark by using the leasing of excess resources up to 90%.

Nested virtualization is a method for migrating instances from the highest to the lowest variable price. Jia et al in [43] introduce Smart Spot instances run on traditional Spot instances. Smart Spot instances work as master instances contain multiple virtual machines. The centralized model is control and monitoring VMs. The master instance determines the optimal cost for transporting the container. Master instances migrate VMs to a new instance when the price changed. The proposed solution lacks the inclusion of the risk of interruption in the transfer decision. Moreover, Smart Spot instances need that all virtual machines instances are the same in size and specification. Besides, Sharma et al. [44] proposed another framework called Spotcheck. Spotcheck relies on the idea of using both excess and on-demand resources to achieve a greater percentage of ensuring the process is not affected by the interruption. Spotcheck uses nested virtualization. Spotcheck migrates the VM's that are on spot leasing to on-demand leasing when price becomes high. Then return migrate to spot leasing when the price decreased. Spotcheck run on the IaaS platform. The obtained results showed that the availability of the system equal to 99.9989% while cost decreased by 80% compared to on-demand VMs. Spotcheck has more total cost than other proposed frameworks. That because using on-demand resources.

B. Cloud-based distributed computing systems to Alleviation interruptions

Many frameworks suggest controlling the spot instance by moving or revoke it based on price fluctuations to get the lowest price and greatest efficiency. Ciavotta et al. [45] provide a new approach via a tool D-SPACE4Cloud. D-SPACE4Cloud can advise cloud computing users when using Hadoop in distributed computing. D-SPACE4Cloud overcome the problem of virtual machines size diversity. The Hadoop work is affected by the different devices associated. Researchers attempted to solve system malfunction problems by using YARN scheduling. Moreover, D-SPACE4Cloud

Authorized licensed use limited to: BOURNEMOUTH UNIVERSITY. Downloaded on June 27,2021 at 02:07:12 UTC from IEEE Xplore. Restrictions apply.

measures time to finish jobs through queueing network (QN) models. D-SPACE4Cloud applied on real systems and the accuracy of results was not less than 30% in the worst cases. The researchers also demonstrated that the use of several small virtual machines (VMs) is better than the large fully equipped, where the small VMs is faster and at a lower cost. In the same context, Bricklayer framework [46] selects a group of smaller instances with a lower price to configure the same requirement of the device required to operate. Bricklayer saved up to 54% of the spot price and 95% of the price on-demand. Optispot framework in [47] gives suggestions based on the number of resources that need to be leasing. Optispot provides a way to link applications with the leased resource and choosing the best spot bid price for keeping the performance of the system is acceptable. Optispot algorithm depends on combining the option to wait until price decrease. The framework relies on the Markov chain that performs a mathematical operation to move from one state to another according to the specified probability rules. This method gives the rule to choose the lowest-cost server. Optispot runs nonlinear programming. The researchers pointed to the lack of the project to apply to a real system to identify the best results. Besides, the solution does not include trigger allocation, deallocation, migration, and replication actions in its framework suggestion. Moreover, Binning et al in [48] present Spotagres which is Parallel Data-Processing Engines (PDEs) model applied to analyze Big Data in clusters. Spotagres consists of two steps. First, providing advice to determine the optimal bid price and the maximal budget for reserving extra resources. The second step is to define fault-tolerance schemes, such as the average meantime between failures (MTBF) that can influence the decision to allocate extra resources. Spotagres determines the higher lease price by determining the level of fault tolerance. According to researchers, the system was able to execute the queries continuously even with a high failure rate. Experimentally, Spotagres was able to provide the same level of queries in the traditional PDE. Spotagres reduced the cost by 99% compared to on-demand costs. Otherwise, Spotagres is limited to measuring prices and providing advice in one region regardless of the data transmission price.

Researchers have provided containers that set the best price and transfer themselves to it. Spoton [49] is a batch computing services run in Linux containers. Spoton uses a checkpointing and replication to avoid fault-tolerance when interrupting the cloud resources. Spoton chooses the best spot price between different availability zones and regions. Spoton supports the parallel job by using a duplicate virtual machine. Spoton decreases the cost by 91.9% as compared to the on-demand price. Spoton performs three steps after determining the best price. First, migrates the tasks before interrupting the cloud resources. The second step creates a checkpoint in memory and local disk. Finally, duplicating the VM on more than one instance to be backup. Spoton offers to migrate VM that quickly can be a move to another server. However, Spoton interrupts existing VMs and creates a new copy in lower-priced resources instead of migrating VMs. Moreover, Spoton's decision is focused on the price for one server type and not choosing other types from hundreds of servers from Spot lease. Shastri & Irwin in [50] introduced a container model named hotspot. Hotspot is a light Linux container with applications to implement. The applications in the resource container are executed. Hotspot container is migrate based on the lowest cost between the excess resources offers. Hotspot is a self-running container that determines the best price then move themself to new leasing without the need for a complete infrastructure. Hotspot reduced the cost by 25% when using it with the extra resources compared to the on-demand leasing. otherwise, Hotspot works on a single server and not support the parallel jobs in Distributed computing systems.

C. A cloud-based framework for machine learning and application workloads

There are different analyze of Big Data such as analyze data related to deep learning. Requires virtual machines have special specifications to analyze that data. Lee & Son in [51] were interested in building distributed computing by using the extra resources to apply deep learning. Deep learning is a mechanism for processing non-linear Big Data to simulate human brain processes. Therefore, deep learning requires devices characterized by an excellent ability to process data and have a graphics processing unit (GPU). Researchers pointed out that the devices with the graphics processing unit usually more expensive in leasing excess resources and less available. The researchers presented a tool called Deepspotcloud. Deepspotcloud creates a pool with the cheapest instances that include GPU. The instances are connected from different regions. Some regions are cheaper and more stable than others. Researchers have proven that virtual machines can be moved between different regions in less than two minutes using the checkpoint method. Deepspotcloud reduces cost by 13% compared to an interruption-based scheduling policy. But the research relied on the migrate virtual machines with a small size only between regions. Therefore, it is difficult to apply this tool to large-size virtual machines because it needs more time to migrate than the allowed time of two minutes.

Table 1 shows a summary of the related work, showing the proposed Framework, the Based-on system, what cloud service has been applied, brief characteristics of the tool and the results according to the research.

TABLE 1:SUMMARY OF THE RELATED WORKS

Reference	Key Points	• Outcomes
[37]	• Develops both batch job and Streaming data • Improves efficiency by choosing the best bid price	• reduced the cost up to 90% compared on-demand. • running time increased by 2%.
[38]	• Deal with Hadoop, Spark, MPI, and BOINC • Use server portfolio to choose the best bid price • Simple coding than other	• Reduce 80% of the total cost
[39]	• Predict price using LSTM	• decrease cost of up to 83.8%
[40]	• Dynamically sizing resources according to predicted requirements • Duplicate resources to overcome inaccurate demand predict	• Reduces formation range 42%, average application complete-time 40% • Increases batch usage 55%

Authorized licensed use limited to: BOURNEMOUTH UNIVERSITY. Downloaded on June 27,2021 at 02:07:12 UTC from IEEE Xplore. Restrictions apply.

[41]	• Using Serverless products to develop the Spark system	• Reduces cost by 21% with 40% improvement in execution time.
[42]	• Combines management of the network buffer and RDMA on the network card	• Improves operations at Spark by 40% using a legacy network stack
[43]	• Centralized model control and monitoring nested VMs.	• Saves up to 15% compared to static instances
[44]	• Use excess & on-demand resources to ensure the process is not affected by the interruption	• System availability equal to 99.9989% • Decrease cost by 80% compared to on-demand
[45]	• Use YARN scheduling. • Ability to handle various virtual machine sizes	• The accuracy of the results is more than 30% in the worst case
[46]	• Selects a group of smaller instances with lower price to configure the same requirements of the hardware required	• Saved up to 54% of the spot price • Reduce cost 95% compared to on-demand
[47]	• The algorithm relies on Markov chain to wait until price decrease	• Markov chain optimize the bid price in varying users and overbid time cap
[48]	• leasing decision depends on choosing the optimal price for spot leasing and fault tolerance mechanism	• Reduced cost to 1% compared to on-demand
[49]	• Spoton is a batch computing service • Spoton duplicate virtual machine to support the parallel job	• Decreases 91.9% of the on-demand cost
[50]	• Migrate VM based on the lowest cost	• Reduced cost to 25%
[51]	• choosing the optimal price instance with dedicated GPU from excess resources across different regions	• Reduces cost by 13% compared to the interruption-based scheduling policy

VI. Counclusion

Through this paper, the main types of distributed computing such as Hadoop and Spark and the main differences between them were clarified. It was also concluded that the Spark system is considered faster than the Hadoop system due to its different data processing characteristics. After that, the advantages of a distributed computing system and the negatives associated with this system are covered in brief. The results of the proposed frameworks for using cloud computing resources to implement a distributed computing system were investigated. The results show that the use of increased cloud resources saves the total cost of distributed computing.

References

[1] M. van Steen and A. S. Tanenbaum, "A brief introduction to distributed systems," Computing, vol. 98, p. 967–1009, 2016.

[2] "Apache Hadoop," 1 10 2020. [Online]. Available: https://hadoop.apache.org/.

[3] I. El Alaoui, Y. Gahi, R. Messoussi, A. Todoskoff and A. Kobi, "Big Data analytics: A comparison of tools and applications," in Proceedings of the Mediterranean Symposium on Smart City Applications, 2017.

[4] J. Dean and S. Ghemawat, "MapReduce: simplified data processing on large clusters," Communications of the ACM, vol. 51, p. 107–113, 2008.

[5] T. White, Hadoop: The definitive guide, " O'Reilly Media, Inc.", 2012.

[6] B. J. Mathiya and V. L. Desai, "Apache hadoop yarn parameter configuration challenges and optimization," in 2015 International Conference on Soft-Computing and Networks Security (ICSNS), IEEE, 2015, pp. 1-6.

[7] T. Condie, N. Conway, P. Alvaro, J. M. Hellerstein, K. Elmeleegy and R. Sears, "MapReduce online.," in Nsdi, 2010.

[8] C. Smith and A. Albarghouthi, "MapReduce program synthesis," Acm Sigplan Notices, vol. 51, p. 326–340, 2016.

[9] V. K. Vavilapalli, A. C. Murthy, C. Douglas, S. Agarwal, M. Konar, R. Evans, T. Graves, J. Lowe, H. Shah, S. Seth and others, "Apache hadoop yarn: Yet another resource negotiator," in Proceedings of the 4th annual Symposium on Cloud Computing, 2013.

[10] K.-H. Lee, Y.-J. Lee, H. Choi, Y. D. Chung and B. Moon, "Parallel data processing with MapReduce: a survey," ACM SIGMOD Record, vol. 40, p. 11–20, 2012.

[11] Y. Yao, J. Wang, B. Sheng, J. Lin and N. Mi, "Haste: Hadoop yarn scheduling based on task-dependency and resource-demand," in 2014 IEEE 7th International Conference on Cloud Computing, 2014.

[12] D. Borthakur, J. Gray, J. S. Sarma, K. Muthukkaruppan, N. Spiegelberg, H. Kuang, K. Ranganathan, D. Molkov, A. Menon, S. Rash and others, "Apache hadoop goes realtime at facebook," in Proceedings of the 2011 ACM SIGMOD International Conference on Management of data, 2011.

[13] M. Schwarzkopf, A. Konwinski, M. Abd-El-Malek and J. Wilkes, "Omega: flexible, scalable schedulers for large compute clusters," in Proceedings of the 8th ACM European Conference on Computer Systems, 2013.

[14] B. Hindman, A. Konwinski, M. Zaharia, A. Ghodsi, A. D. Joseph, R. H. Katz, S. Shenker and I. Stoica, "Mesos: A platform for fine-grained resource sharing in the data center.," in NSDI, 2011.

[15] R. Ramakrishnan, B. Sridharan, J. R. Douceur, P. Kasturi, B. Krishnamachari-Sampath, K. Krishnamoorthy, P. Li, M. Manu, S. Michaylov, R. Ramos and others, "Azure data lake store: a hyperscale distributed file service for big data analytics," in Proceedings of the 2017 ACM International Conference on Management of Data, 2017.

[16] "Apache Spark," 1 10 2020. [Online]. Available: https://spark.apache.org/.

[17] S. Salloum, R. Dautov, X. Chen, P. X. Peng and J. Z. Huang, "Big data analytics on Apache Spark," International Journal of Data Science and Analytics, vol. 1, p. 145–164, 2016.

[18] I. Mavridis and H. Karatza, "Performance evaluation of cloud-based log file analysis with Apache Hadoop and Apache Spark," Journal of Systems and Software, vol. 125, p. 133–151, 2017.

[19] M. Zaharia, M. Chowdhury, T. Das, A. Dave, J. Ma, M. Mccauley, M. Franklin, S. Shenker and I. Stoica, "Fast and interactive analytics over Hadoop data with Spark," Usenix Login, vol. 37, p. 45–51, 2012.

[20] M. M. Rathore, H. Son, A. Ahmad, A. Paul and G. Jeon, "Real-time big data stream processing using GPU with spark over hadoop ecosystem," International Journal of Parallel Programming, vol. 46, p. 630–646, 2018.

[21] H. M. Makrani and H. Homayoun, "Memory requirements of hadoop, spark, and MPI based big data applications on commodity server class architectures," in 2017 IEEE International Symposium on Workload Characterization (IISWC), 2017.

[22] W. Huang, L. Meng, D. Zhang and W. Zhang, "In-memory parallel processing of massive remotely sensed data using an apache spark on hadoop yarn model," IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, vol. 10, p. 3–19, 2016.

[23] Y. Samadi, M. Zbakh and C. Tadonki, "Comparative study between Hadoop and Spark based on Hibench benchmarks," in 2016 2nd International Conference on Cloud Computing Technologies and Applications (CloudTech), 2016.

[24] A. V. Hazarika, G. J. S. R. Ram and E. Jain, "Performance comparision of Hadoop and spark engine," in 2017 International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud)(I-SMAC), 2017.

[25] M. A. Hedjazi, I. Kourbane, Y. Genc and B. Ali, "A comparison of Hadoop, Spark and Storm for the task of large scale image

Authorized licensed use limited to: BOURNEMOUTH UNIVERSITY. Downloaded on June 27,2021 at 02:07:12 UTC from IEEE Xplore. Restrictions apply.

classification," in 2018 26th Signal Processing and Communications Applications Conference (SIU), 2018.

[26] V. D. Thoke and V. Sangli, "Theory of distributed computing and parallel processing with its applications, advantages and disadvantages.," International Journal of Innovation in Engineering, Research and Technology , 2014.

[27] V. A. Pavskiy, K. V. Pavskiy and A. A. Paznikov, "Mathematical models and calculation of reliability indices of scalable distributed computer systems under full restoration," in 2018 XIV International Scientific-Technical Conference on Actual Problems of Electronics Instrument Engineering (APEIE), 2018.

[28] O. Wennergren, M. Vidhall and J. Sörensen, Transparency analysis of distributed file systems: With a focus on interplanetary file system, Sweden, School of Information Technology at the University of Skövde, 2018.

[29] J. Pourqasem, S. Karimi and S. A. Edalatpanah, "Comparison of cloud and grid computing," American Journal of Software Engineering, vol. 2, p. 8–12, 2014.

[30] D. A. Prathibha, B. Latha and G. Sumathi, "Issues in adapting cluster, grid and cloud computing for HPC applications," International Journal of Conceptions on Computing and Information Technology, vol. 2, p. 12–16, 2014.

[31] E. R. Kaur, "A review of computing technologies: distributed, utility, cluster, grid and cloud computing," International Journal of Advanced Research in Computer Science and Software Engineering, vol. 5, p. 144–148, 2015.

[32] K. Kaur and A. K. Rai, "A comparative analysis: Grid, cluster and cloud computing," International Journal of Advanced Research in Computer and Communication Engineering, vol. 3, p. 5730–5734, 2014.

[33] H. K. Ala'a Al-Shaikh, A. Sharieh and A. Sleit, "Resource utilization in cloud computing as an optimization problem," International Journal of Advanced Computer Science and Applications,, vol. 7, no. 6, pp. 336-342, 2016.

[34] N. A. Al Etawi, "A comparison between cluster, grid, and cloud computing," International Journal of Computer Applications, vol. 179, p. 37–42, 2018.

[35] S. Schäffner, "Data Management in Distributed Systems," Chair of Network Architectures and Services, pp. 113-120, 2018.

[36] J. Edinger, "Context-aware task scheduling in distributed computing systems," 29 April 2019. [Online]. Available: https://madoc.bib.uni-mannheim.de/51051.

[37] P. Sharma, T. Guo, X. He, D. Irwin and P. Shenoy, "Flint: Batch-interactive data-intensive processing on transient servers," in Proceedings of the Eleventh European Conference on Computer Systems, 2016.

[38] P. Sharma, D. Irwin and P. Shenoy, "Portfolio-driven resource management for transient cloud servers," Proceedings of the ACM on Measurement and Analysis of Computing Systems, vol. 1, p. 1–23, 2017.

[39] F. Xu, H. Zheng, H. Jiang, W. Shao, H. Liu and Z. Zhou, "Cost-Effective Cloud Server Provisioning for Predictable Performance of Big Data Analytics," IEEE Transactions on Parallel and Distributed Systems, vol. 30, pp. 1036-1051, 2019.

[40] L. Liu and H. Xu, "Elasecutor: Elastic executor scheduling in data analytics systems," in Proceedings of the ACM Symposium on Cloud Computing, 2018.

[41] A. Jain, "SplitServe: Efficiently splitting complex workloads over IaaS and FaaS," 2019.

[42] H. Li, T. Chen and W. Xu, "Improving spark performance with zero-copy buffer management and RDMA," in 2016 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), 2016.

[43] Q. Jia, Z. Shen, W. Song, R. van Renesse and H. Weatherspoon, "Smart spot instances for the supercloud," in Proceedings of the 3rd Workshop on CrossCloud Infrastructures & Platforms, 2016.

[44] P. Sharma, S. Lee, T. Guo, D. Irwin and P. Shenoy, "Spotcheck: Designing a derivative iaas cloud on the spot market," in Proceedings of the Tenth European Conference on Computer Systems, 2015.

[45] M. Ciavotta, E. Gianniti and D. Ardagna, "D-space4cloud: a design tool for big data applications," in International Conference on Algorithms and Architectures for Parallel Processing, 2016.

[46] W. Wong, L. Corneo, A. Zavodovski, P. Zhou, N. Mohan and J. Kangasharju, "Bricklayer: Resource Composition on the Spot Market," in ICC 2020-2020 IEEE International Conference on Communications (ICC), 2020.

[47] D. J. Dubois and G. Casale, "OptiSpot: minimizing application deployment cost using spot cloud resources," Cluster Computing, vol. 19, p. 893–909, 2016.

[48] C. Binnig, A. Salama, E. Zamanian, M. El-Hindi, S. Feil and T. Ziegler, "Spotgres-parallel data analytics on spot instances," in 2015 31st IEEE International Conference on Data Engineering Workshops, 2015.

[49] S. Subramanya, T. Guo, P. Sharma, D. Irwin and P. Shenoy, "Spoton: a batch computing service for the spot market," in Proceedings of the sixth ACM symposium on cloud computing, 2015.

[50] S. Shastri and D. Irwin, "HotSpot: automated server hopping in cloud spot markets," in Proceedings of the 2017 Symposium on Cloud Computing, 2017.

[51] K. Lee and M. Son, "Deepspotcloud: leveraging cross-region gpu spot instances for deep learning," in 2017 IEEE 10th International Conference on Cloud Computing (CLOUD), 2017.

Authorized licensed use limited to: BOURNEMOUTH UNIVERSITY. Downloaded on June 27,2021 at 02:07:12 UTC from IEEE Xplore. Restrictions apply.