International Journal for Innovative Engineering and Management Research

A Peer Revieved Open Access International Journal

www.ijiemr.org

IJIEMR Transactions, online available on 18th Feb 2021. Link

:http://www.ijiemr.org/downloads.php?vol=Volume-10&issue=ISSUE-02

DOI: 10.48047/IJIEMR/V10/I02/13

Title **LOCALITY-AWARE SCHEDULING FOR CONTAINERS IN CLOUD COMPUTING**

Volume 10, Issue 02, Pages: 64-71.

Paper Authors

Smitha Nayak

International Journal for Innovative Engineering and Management Research
A Peer Revieved Open Access International Journal
www.ijiemr.org

LOCALITY-AWARE SCHEDULING FOR CONTAINERS IN CLOUD COMPUTING

Smitha Nayak[1]
Assistant professor, Research Scholar[1]
Shri Jagdish prasad Jhabarmal Tibrewala University[1],
Smithank@gmail.com[1]

ABSTRACT

The state-of-the-art scheduler of containerized cloud services considers load balance as the only criterion; many other important properties, including application performance, are overlooked. In the era of Big Data, however, applications evolve to be increasingly more data-intensive thus perform poorly when deployed on containerized cloud services. To that end, this paper aims to improve today's cloud service by taking application performance into account for the next-generation containers. More specifically, in this work we build and analyze a new model that respects both load balance and application performance. Unlike prior studies, our model abstracts the dilemma between load balance and application performance into a unified optimization problem and then employs a statistical method to efficiently solve it. The most challenging part is that some sub-problems are extremely complex (for example, NP-hard), and heuristic algorithms have to be devised. Last but not least, we implement a system prototype of the proposed scheduling strategy for containerized cloud services. Experimental results show that our system can significantly boost application performance while preserving relatively high load balance.

SCOPE OF WORK

The state-of-the-art scheduler of containerized cloud services considers load balance as the only criterion; many other important properties, including application performance, are overlooked. In the era of Big Data, however, applications evolve to be increasingly more data-intensive thus perform poorly when deployed on containerized cloud services. To that end, this paper aims to improve today's cloud service by taking application performance into account for the next-generation containers. More specifically, in this work we build and analyze a new model that respects both load balance and application performance. Unlike prior studies, our model abstracts the dilemma between load balance and application performance into a unified optimization problem and then employs a statistical method to efficiently solve it. The most challenging part is that some sub-problems are extremely complex (for example, NP-hard), and heuristic algorithms have to be devised. Last but not least, we implement a system prototype of the proposed scheduling strategy for containerized cloud services. Experimental results show that our system can significantly boost application performance while preserving relatively high load balance.

The past few years have seen a growing number of mobile and sensor applications that rely on Cloud support. The role of the Cloud is to allow these resource-limited devices to offload and execute some of their compute-intensive tasks in the Cloud for energy saving and/or faster processing. However, such offloading to the Cloud may result in high network overhead which is not suitable for many mobile/sensor applications that require low latency. In this paper, we propose a locality-aware load sharing technique that allows edge resources to share their workload in order to maintain the low latency requirement of Mobile-Cloud applications. Specifically, we study how to determine which edge nodes should be used to share the workload with and how much of the workload should be shared to each node. Our experiments show that our locality-aware load sharing technique is able to maintain low average end-to-end latency of mobile applications with low latency variation, while achieving good utilization of resources in the presence of a dynamic workload.

OBJECTIVES OF THE STUDY

1. To study predictive analysis of locality-aware storagetier data blocks over hadoop
2. To design locality-aware load sharing in mobile cloud computing
3. To design toward locality-aware scheduling for containerized cloud services
4. To study locality-aware scheduling for containers in cloud computing

LITERATURE REVIEW

- **LaValle, Steve, et al (2011)** The term 'Big Data analytics' refers to a large-scale solution for managing giant datasets in a parallel environment. Hadoop is an ecosystem that processes large datasets in distributed computing scenario. The ecosystem is further categorized into four sub-projects i.e. HDFS, MapReduce, YARN and Hadoop Commons. The Hadoop Distributed File System (HDFS) is a backbone of ecosystem, which helps storing and processing large datasets. Recently, HDFS is upgraded to heterogeneous storage-tier environment that cope with data block processing over multiple storage devices i.e. DISK, SSD and RAM. The block placement policy dispatches data blocks to the devices without calculating I/O transfer parameters and locality perspectives. Moreover, HDFS selects random Datanodes that could be located into the next rack having longer path than local rack. This increases the data block processing latency and results in a huge delay for replica management in heterogeneous storage-tier. To resolve this issue, we propose a predictive analysis that build a locality-aware storage-tier node summary and predict the most nearby available storage-tier for block job processing. The experimental evaluation depicts that the proposed approach reduces data block transfer time overhead, replica transfer time overhead and decreases node paths to an optimal accessibility over the cluster.

- **Cloudera (2016)** In this paper, we propose a framework for privacy-preserving outsourced drug discovery in the cloud, which we refer to as POD. Specifically, POD is designed to allow the cloud to securely use multiple drug formula providers' drug formulas to train Support Vector Machine (SVM) provided by the analytical model provider. In our approach, we design secure computation protocols to allow the cloud server to perform commonly used integer and fraction computations. To securely train the SVM, we design a secure SVM parameter selection protocol to select two SVM parameters and construct a secure sequential minimal optimization protocol to privately refresh both selected SVM parameters. The trained SVM classifier can be used to determine whether a drug chemical compound is active or not in a privacy-preserving way. Lastly, we prove that the proposed POD achieves the goal of SVM training and chemical compound classification without privacy leakage to unauthorized parties, as well as demonstrating its utility and efficiency using three real-world drug datasets.
- **A. Kala Karun and K. Chitharanjan (2013)** Cloud-assisted Internet of Things (IoT) provides a promising solution to data booming problems for the ability constraints of individual objects. However, with the leverage of cloud, IoT faces new security challenges for data mutuality between two parties, which is introduced for the first time in this paper and not currently addressed by traditional approaches. We investigate a secure cloud-assisted IoT data managing method to keep data confidentiality when collecting, storing and accessing IoT data with the assistance of a cloud with the consideration of users increment. The proposed system novelly applies a proxy re-encryption scheme, which was proposed in \cite{XJW15}. Hence, a secure IoT under our proposed method could resist most attacks from both insiders and outsiders of IoT to break data confidentiality, and meanwhile with constant communication cost for re-encrytion anti incremental scale of IoT. We further show the method is practical by numerical results.
- **Abbas, A., Wu, Z., Siddiqui, I. F., & Lee, S. U. J. (2016)** Information leakage is the inadvertent disclosure

International Journal for Innovative Engineering and Management Research
A Peer Revieved Open Access International Journal
www.ijiemr.org

of sensitive information through correlation of records from several databases/collections of a cloud data warehouse. Malicious insiders pose a serious threat to cloud data security and this justifies the focus on information leakage due to rogue employees or to outsiders using the credentials of legitimate employees. The discussion in this paper is restricted to NoSQL databases with a flexible schema. Data encryption can reduce information leakage, but it is impractical to encrypt large databases and/or all fields of database documents. Encryption limits the operations that can be carried on the data in a database. It is thus, critical to distinguish sensitive documents in a data warehouse and concentrate on efforts to protect them. The capacity of a leakage channel introduced in this work quantifies the intuitively obvious means to trigger alarms when an insider attacker uses excessive computer resources to correlate information in multiple databases. The Sensitivity Analysis based on Data Sampling (SADS) introduced in this paper balances the trade-offs between higher efficiency in identifying the risks posed by information leakage and the accuracy of the results obtained by sampling very large collections of documents. The paper reports on experiments assessing the effectiveness of SADS and the use of selective disinformation to limit information leakage. Cloud services identifying sensitive records and reducing the risk of information leakage are also discussed.

- **Y. Tsuruoka (2016)** With the advent of cloud computing, more and more people tend to outsource their data to the cloud. As a fundamental data utilization, secure keyword search over encrypted cloud data has attracted the interest of many researchers recently. However, most of existing researches are based on an ideal assumption that the cloud server is "curious but honest", where the search results are not verified. In this paper, we consider a more challenging model, where the cloud server would probably behave dishonestly. Based on this model, we explore the problem of result verification for the secure ranked keyword search. Different from previous data verification schemes, we propose a novel deterrent-based scheme. With our carefully devised verification data, the cloud server cannot know which data owners, or how many data owners exchange anchor data which will be used for verifying the cloud server's misbehavior. With our systematically designed verification construction, the cloud server cannot know which data owners' data are embedded in the verification data buffer, or how

International Journal for Innovative Engineering and Management Research
A Peer Revieved Open Access International Journal
www.ijiemr.org

many data owners' verification data are actually used for verification. All the cloud server knows is that, once he behaves dishonestly, he would be discovered with a high probability, and punished seriously once discovered. Furthermore, we propose to optimize the value of parameters used in the construction of the secret verification data buffer.

METHODOLOGY

The predictive analysis consists of two phases i.e. (i) Storage-tier summary container and (ii) Predict the most nearby Data node.

The storage-tier summary container collects all the Data node and storage media information i.e. computing capacity and storage-tier devices with volume statistics. Moreover, the media predictor performs training sessions over the dataset and predicts the most nearby Data node with available storage-tier media as seen from Figure.

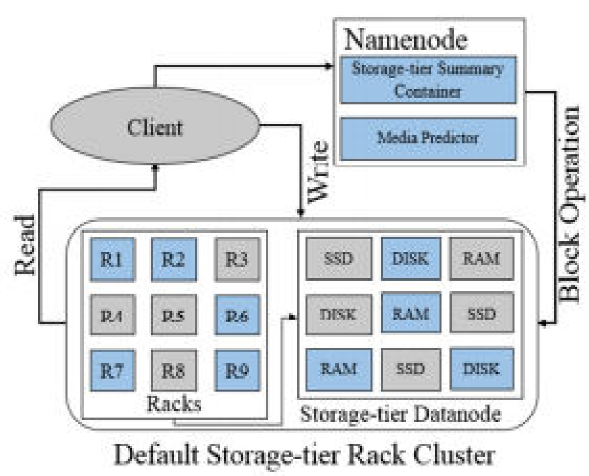


Figure Storage-tier Predictive Analysis over HDFS Cluster

Storage-tier summary container

The summary container consists of Data node information i.e. CPU, storage medias, accessibility time, 1 MB data block receiving time and volume sizes of each storage.

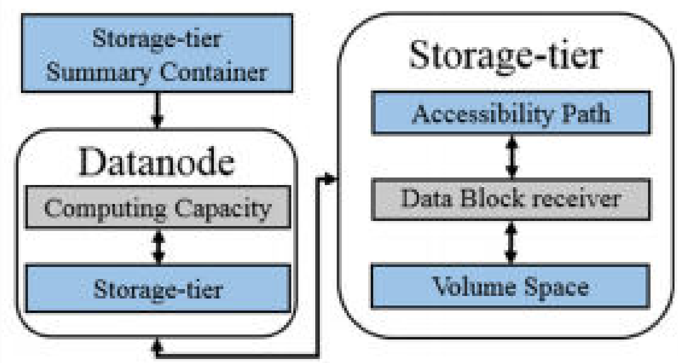


Figure Storage-tier summary Architecture

RESULTS

Environment

The ecosystem consists of Intel Xeon processor with 8 CPUs, 32GB memory and storage devices i.e. 1TB Hard disk drive and 128GB Samsung SSD. In addition to that, we use Intel core i5 with 4 Core, 16GB memory and storage devices i.e. 1TB Hard disk drive and 128 GB Samsung SSD. We install 5 virtual machines having virtual box 5.0.16 as seen from Table.

The experimental dataset consists of:

(i) 250 random SSD word count data blocks of 64MB (40GB size),

(ii) 250 random DISK word count data blocks (40GB size) and

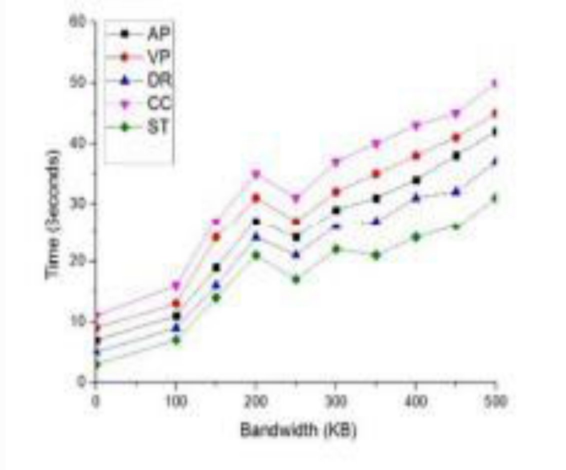
(iii) 250 random RAM word count data blocks (40GB size).

Storage-tier summary collector

The collector fetches event traces of computing capacity, storage-tier I/O,

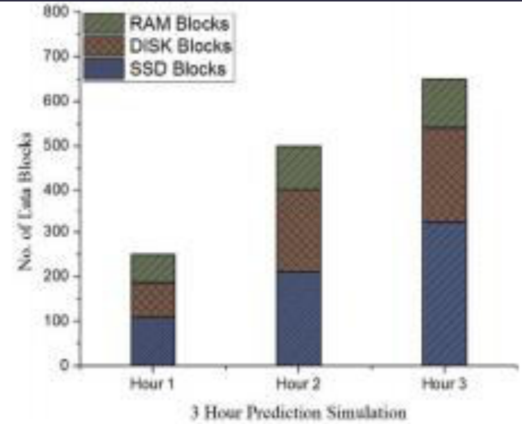
International Journal for Innovative Engineering and Management Research
A Peer Revieved Open Access International Journal
www.ijiemr.org

Accessibility Path time stamp, Data block receiver timestamp and Volume Space statistics over container. The message length varies between 0.5 ≤ size ≥ 5 KB and consumes a resource between 0.2 ≤ Bandwidth ≥ 500 KB/s. The summary container stores 2.7 GB of log information over 120GB data blocks as observed from Figure.

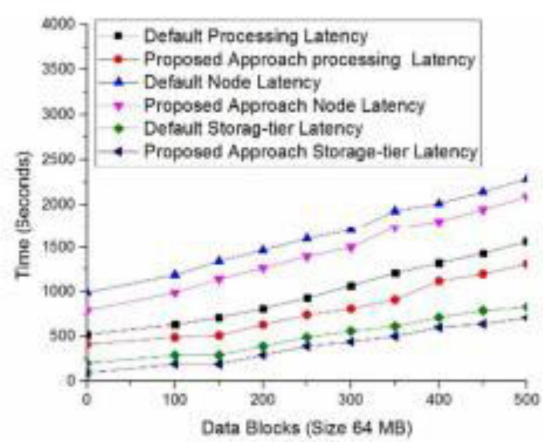


Graph Storage-tier Summary Container Message Collector

After generating container messages, we perform prediction simulations over three '250' random data blocks. In the first hour of simulation, we observe that predictor detects pattern of '109' SSD data blocks, '78' DISK data blocks and '63' RAM data blocks. In the second hour of simulation, we use '500' random data blocks and analyze that predictor observes pattern of '211' SSD data blocks, '192' DISK data blocks and '97' RAM data blocks. In the third hour of simulation, we evaluate '750' random data blocks and evaluate that predictor observes pattern of '322' SSD data block, '219' DISK data blocks and '109' RAM data blocks as observed from Figure.



Graph Storage-tier media block job prediction



Graph Latency optimization over HDFS Cluster

The media predictor depicts locality-aware nearby Data node statistics with available functional media. The predictor lists processing timestamp, accessibility timestamp and completion timestamp of data blocks over respective storage media. As a result, we calculate locality-aware path and observes that proposed processing, node and storage-tier approaches are 39.1%, 54.7% and 22.9% efficient than default data block processing. This reduces storage-tier latency, node latency and overall processing latency as observed from Figure

CONCLUSION

In this paper, we study the problem of load sharing to handle runtime dynamics in a Mobile-Edge Computing (MEC) environment. Our motivation is based on the dynamic property of the workload in MEC along with the low latency requirement for many of today's mobile/IoT applications. The Edge Cloud platform that has been proposed to provide computational ooading support for mobile applications faces additional challenges in handling workload dynamics since the nodes in Edge Clouds are typically connected by WAN with high network latency and limited bandwidth. We propose a locality-aware load sharing technique that allows edge nodes to share their workload to other nodes to meet the low latency requirement of the mobile applications in the case of workload increases. Our load sharing technique allows nodes to

1) Intelligently determines whether to share their workload to other nodes,
2) Selectively chooses which nodes the workload should be shared with, and
3) Determines how much of the workload should be shared. Our experimental results based on a real Twitter's trace show that our locality-aware load sharing technique is able to keep the overall latency of mobile applications close to the applications' desired goals as well as better utilize resources even in the case of dynamic workload.

REFERENCES

[1] LaValle, Steve, et al. "Big data, analytics and the path from insights to value." MIT sloan management review 52.2, 2011, pp. 21.

[2] "Welcome to Apache™ Hadoop®!" 2014. [Online]. Available: http://hadoop.apache.org/. Accessed: Mar. 13, 2017.

[3] M. Technologies, "Featured customers", 2016. [Online]. Available: https://www.mapr.com/. Accessed: Mar. 13, 2017.

[4] Cloudera, "The modern platform for data management and analytics," Cloudera, 2016. [Online]. Available: http://www.cloudera.com/. Accessed: Mar. 13, 2017.

[5] "Apache Hadoop 2.7.2 – Apache Hadoop YARN," 2016. [Online]. Available: https://hadoop.apache.org/docs/r2.7.2/hadoopyarn/hadoop-yarn-site/YARN.html. Accessed: Mar. 13, 2017.

[6] "Apache Hadoop 2.7.2 – MapReduce Tutorial," 2016. [Online]. Available: https://hadoop.apache.org/docs/stable/hadoopmapreduce-client/hadoop-mapreduce-clientcore/MapReduceTutorial.html. Accessed: Mar. 13, 2017.

[7] "Apache Hadoop 2.7.2 – HDFS users guide," 2016. [Online]. Available: https://hadoop.apache.org/docs/stable/hadoopproject-dist/hadoop hdfs/HdfsUserGuide.html. Accessed: Mar. 13, 2017.

[8] A. Kala Karun and K. Chitharanjan, "A review on Hadoop — HDFS infrastructure extensions," 2013 IEEE CONFERENCE

ON INFORMATION AND COMMUNICATION TECHNOLOGIES, Apr. 2013.

[9] Abbas, A., Wu, Z., Siddiqui, I. F., & Lee, S. U. J. (2016). An approach for optimized feature selection in software product lines using unionfind and Genetic Algorithms. Indian Journal of Science and Technology, 9(17)

[10] "Apache Hadoop 2.7.2 – HDFS storage-tier," 2016. [Online]. Available: https://hadoop.apache.org/docs/r2.7.3/hadoop-project-dist/hadoop
hdfs/ArchivalStorage.html
Accessed: Mar. 13, 2017.

[11] Abbas, A., Siddiqui, I. F., & Lee, S. U. J. (2016). Multi-Objective Optimization of Feature Model in Software Product Line: Perspectives and Challenges. Indian Journal of Science and Technology, 9(45).

[12] Y. Tsuruoka, "Cloud computing - current status and future directions," Journal of Information Processing, vol. 24, no. 2, 2016, pp. 183–194.

[13] ABBAS, A., SIDDIQUI, I. F., & LEE, S. U. J. (2017). CONTEXTUAL VARIABILITY MANAGEMENT OF IOT APPLICATION WITH XML-BASED FEATURE MODELLING. Journal of Theoretical & Applied Information Technology, 95(6).

[14] C. Rodríguez-Quintana, A. F. Díaz, J. Ortega, R. H. Palacios, and A. Ortiz, "A new Scalable approach for distributed Metadata in HPC," in Algorithms and Architectures for Parallel Processing. Springer Nature, 2016, pp. 106-117.

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