SCALABLE AND ADAPTIVE DATA REPLICA PLACEMENT FOR GEO DISTRIBUTED CLOUD STORAGES

[1]NARALA MANISH KUMAR, [2]SAIKRISHNA.D
[1]MCA Student, [2]Assistant Professor
DEPARTMENT OF MCA
SREE CHAITANYA COLLEGE OF ENGINEERING, KARIMNAGAR

ABSTRACT

In geo-distributed cloud storage systems, data replication has been widely used to serve the ever more users around the world for high data reliability and availability. How to optimize the data replica placement has become one of the fundamental problems to reduce the inter-node traffic and the system overhead of accessing associated data items. In the big data era, traditional solutions may face the challenges of long running time and large overheads to handle the increasing scale of data items with time-varying user requests. Therefore, novel offline community discovery and online community adjustment schemes are proposed to solve the replica placement problem in a scalable and adaptive way. The offline scheme can find a replica placement solution based on the average read/write rates for a certain period of time. The scalability can be achieved as 1) the computation complexity is linear to the amount of data items and 2) the data-node communities can evolve in parallel for a distributed replica placement. Furthermore, the online scheme is adaptive to handle the bursty data requests, without the need to completely override the existing replica placement. Driven by real-world data traces, extensive performance evaluations demonstrate the effectiveness of our design to handle large-scale datasets.

I. INTRODUCTION

In the current era of big data, geo-distributed cloud storage systems need to manage, manipulate, and analyze a large scale of data for the emerging data-intensive applications. According to the IDC report, the volume of data is doubling every two years and thus will reach a staggering 44 zettabytes by 2020 [1]. To serve the ever more users around the world, data replication among geo-distributed storages has been widely used to increase data reliability and availability [2]. Placing requested data closer to end users helps to lower the user experienced service delay and the inter-node data read traffic, which motivates intensive research about data replica placement. Modern service providers, e.g., Facebook, maintain a full copy of user data in each data center [3]. However, this may generate unnecessarily high inter-node synchronization traffic to maintain consistency among data and replicas. Therefore, the inter-node traffic can be reduced by selecting a proper number of data replicas. Apart from the inter-node traffic, the storage locations of data replicas may also affect the system overhead of accessing associated data items [4], [5]. It is worth noting that users may request multiple data items in one transaction. For example, in online analytical processing (OLAP) systems, a query may be executed by accessing multiple data blocks [6]. The system overhead could be reduced if fewer storage nodes are involved to handle such a request. The reason is that a certain overhead, e.g., the establishment of TCP connections, will be introduced if the read request is dispatched to a storage node. In short, data replica placement reduces the system overhead by placing associated data items together in the same storage location. With the increasing number of data items, how to choose the proper number and storage locations of data replicas becomes a critical issue. Various data replica placement schemes have been proposed to seek optimal data storage locations, which are typically implemented in a centralized/offline way: At every distributed storage node handling the user requests, the data access logs are captured. Then, a central controller is deployed to collect all logs and analyze the request frequency of each data item. The extracted information is fed into the replica placement algorithms, e.g., mathematical programming [8] and graph partitioning [5], [7], [9], which finally output the storage locations of data replicas. These centralized/offline schemes can iteratively approximate the optimal solutions with high accuracy. Although intuitively valid in design, the centralized/offline schemes may meet two practical challenges when applied to a large-scale storage system. First of all, a long running time of the placement scheme is expected when a large amount of data items are deployed at many storage nodes [4]. Furthermore, faced with time-varying data requests, these offline solutions are slow to react to the realtime changes in workloads [10]. For a large scale storage system with user request uncertainties, data replica placement schemes should be 1) highly efficient with small computation overhead for a quick placement decision, and 2) flexible to change the storage locations of data replicas in an online fashion. Therefore, it is imperative to solve the replica placement problem in a more scalable and adaptive way. In this paper, based on the overlapping community discovery and adjustment, we design scalable and adaptive data replica placement schemes in geo-distributed cloud storage systems. A data-node community is defined as the group of a storage node and all data items placed at it, which should have more internal data access requests than external ones. Therefore, a more compact community structure means more data requests are served locally with lower system overhead and less inter-node traffic. Unlike traditional centralized placement schemes, communities can evolve to decide whether each data replica should be placed at the node in a parallel and adaptive way. The scalability of our design can be achieved by this distributed implementation along with the computation complexity linear to the amount of data items. Our major contributions in this paper include: A novel distributed overlapping community discovery scheme is proposed to solve the data replica placement problem in a scalable way. This offline scheme can find a replica placement solution based on the average read/write rates for a certain time period. Guided by the offline scheme, an online community adjustment scheme is proposed to adaptively handle the bursty requests. The worst-case performance guarantees of the proposed schemes are provided via theoretical analysis. Extensive evaluation results driven by real-world data traces show the superiority of our design over the state-of-the-art replica placement methods.

II. SYSTEM ANALYSIS

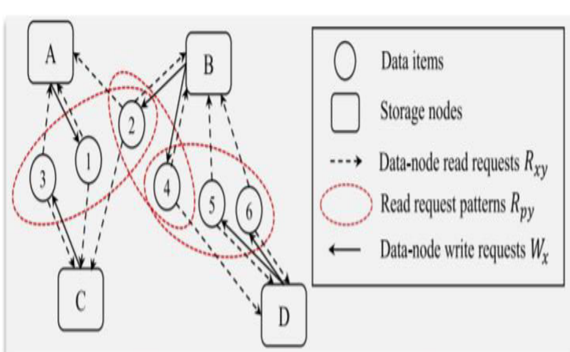
EXISTING SYSTEM

Apart from the inter-node traffic, the storage locations of data replicas may also affect the system overhead of accessing associated data items [4], [5]. It is worth noting that users may request multiple data items in one transaction. For example, in online analytical processing (OLAP) systems, a query may be executed by accessing multiple data blocks [6]. The system overhead could be reduced if fewer storage nodes are involved to handle such a request. The reason is that a certain overhead, e.g., the establishment of TCP connections, will be introduced if the read request is dispatched to a storage node. In short, data replica placement reduces the system overhead by placing associated data items together in the same storage location. With the increasing number of data items, how to choose the proper number and storage locations of data replicas becomes a critical issue. Various data replica placement schemes have been proposed to seek optimal data storage locations, which are typically implemented in a centralized/offline way: At every distributed storage node handling the user requests, the data access logs are captured. Then, a central controller is deployed to collect all logs and analyze the request frequency of each data item.

PROPOSED SYSTEM:

online community adjustment schemes are proposed to solve the replica placement problem in a scalable and adaptive way. The offline scheme can find a replica placement solution based on the average read/write rates for a certain period of time. The scalability can be achieved as 1) the computation complexity is linear to the amount of data items and 2) the data-node communities can evolve in parallel for a distributed replica placement. Furthermore, the online scheme is adaptive to handle the bursty data requests, without the need to completely override the existing replica placement. Driven by real-world data traces, extensive performance evaluations demonstrate the effectiveness of our design to handle large-scale datasets.

SYSTEM ARCHITECTURE:



III. IMPLEMENTATION

Data Ownr Module

In this module, the data owner uploads their data in the cloud server. For the security purpose the data owner encrypts the data file's blocks and then store in the cloud. The data owner can check the replication of the file's blocks over Corresponding cloud server. The Data owner can have capable of manipulating View Files, Upload File, and Update File.

DATA USER

In this module, remote user logs in by using his user name and password. After he will request for secrete key of required file's blocks from cloud servers, and get the secrete key. After getting secrete key he is trying to download file's blocks by entering file's blocks name and secrete key from cloud server and performs the following operations View Service Recommendations, Search, Download, View Files, Search Request, Download Request.

IV. CONCLUSION

Observing the increasing scale of data items and time-varying data requests in geo-distributed storage systems, we proposed scalable and adaptive data replica placement schemes based on the overlapping community discovery approach to improve the efficiency of making placement decisions. With an overall consideration of the inter-node traffic and system overhead of accessing associated data, data-node communities can evolve to decide whether each data replica should be placed at each node in a parallel way. This distributed implementation along with the linear computation complexity over the number of data items ensures the scalability of our design. The online scheme was further proposed to adaptively handle the bursty requests. The worst-case performance bound was also theoretically analyzed. Evaluation driven by real-world datasets showed that compared with the centralized scheme ADP, the proposed scheme DCD incurs similar data access overhead, while greatly reduces the running time. Guided by the offline DCD, the data access overhead can be further reduced by about 30 percent with the online OCA. In future work, more performance metrics, e.g., data access latencies, cost of storage, and load balance among storage nodes, will be considered in the data replica placement. These metrics may also influence the performance of the storage system. Furthermore, a prototype of the geo-distributed cloud storage system based on Amazon EC2 clusters will be built for a series of real-world experiments to validate the performance of the proposed data replica placement schemes.

REFERENCES

[1] Data Growth, Business Opportunities, and the IT Imperatives, 2014. [Online]: https://www.emc.com/leadership/digitaluniverse/2014iview/executive-summary.htm

[2] Y. Mansouri, A. N. Toosi, and R. Buyya, “Data storage management in cloud environments: Taxonomy, survey, and future directions,” ACM Comput. Surv., vol. 50, no. 6, 2017, Art. no. 91.

[3] G. Liu, H. Shen, and H. Chandler, “Selective data replication for online social networks with distributed data centers,” IEEE Trans. Parallel Distrib. Syst., vol. 27, no. 8, pp. 2377–2393, Aug. 2016.

[4] S. Agarwal, J. Dunagan, N. Jain, S. Saroiu, A. Wolman, and H. Bhogan, “Volley: Automated data placement for geo-distributed cloud services,” in Proc. USENIX Conf. Netw. Syst. Des. Implementation, 2010, Art. no. 2.

[5] B. Yu and J. Pan, “A framework of hypergraph-based data placement among geo-distributed datacenters,” IEEE Trans. Services Comput., to be published, doi: 10.1109/TSC.2017.2712773.

[6] Q. Pu et al., “Low latency geo-distributed data analytics,” in Proc. ACM Conf. Special Interest Group Data Commun., 2015, pp. 421–434.

[7] B. Yu and J. Pan, “Sketch-based data placement among geo-distributed datacenters for cloud storages,” in Proc. IEEE INFOCOM, 2016, pp. 1–9.

[8] X. Ren, P. London, J. Ziani, and A. Wierman, “Datum: Managing data purchasing and data placement in a geo-distributed data market,” IEEE/ACM Trans. Netw., vol. 26, no. 2, pp. 893–905, Apr. 2018.

[9] L. Jiao, J. Li, W. Du, and X. Fu, “Multi-objective data placement for multi-cloud socially aware services,” in Proc. IEEE INFOCOM, 2014, pp. 28–36.

[10] A. Charapko, A. Ailijiang, and M. Demirbas, “Adapting to access locality via live data migration in globally distributed datastores,” in Proc. IEEE Int. Conf. Big Data, 2018, pp. 3321–3330.

[11] D. A. Tran, K. Nguyen, and C. Pham, “S-CLONE: Socially-aware data replication for social networks,” Comput. Netw., vol. 56, no. 7, pp. 2001–2013, 2012.

[12] S. Traverso, K. Huguenin, I. Trestian, V. Erramilli, N. Laoutaris, and K. Papagiannaki, “TailGate: Handling long-tail content with a little help from friends,” in Proc. 21st Int. Conf. World Wide Web, 2012, pp. 151–160.

[13] S. Raindel and Y. Birk, “Replicate and bundle (RnB)–A mechanism for relieving bottlenecks in data centers,” in Proc. IEEE 27th Int. Symp. Parallel Distrib. Process., 2013, pp. 601–610. [14] R. Nishtala et al., “Scaling

memcache at Facebook," in Proc. USENIX Conf. Netw. Syst. Des. Implementation, 2013, pp. 385–398.

[15] A. Atrey, G. V. Seghbroeck, H. Mora, F. D. Turcka, and B. Volckaert, "SpeCH: A scalable framework for data placement of data-intensive services in geo-distributed clouds," J. Netw. Comput. Appl., vol. 142, pp. 1–14, 2019.