International Journal of Science Engineering and Advance Technology

A New Efficient Privacy for a Multi-Skyline Queries with Map Reduce

Badavathu Kavitha
Lecturer, Dept. of Computer Science, Sri Durga Malleswara Siddhartha Mahila kalasala,Vijayawada, A.P., India.

Abstract — The skyline query technology has pulled in much consideration as of late. This is for the most part because of the significance of horizon brings about numerous applications, for example, multi-criteria basic leadership, information mining, and data suggested frameworks. The horizon administrator has pulled in impressive consideration as of late because of its wide applications. Be that as it may, registering a horizon is testing today since we need to manage enormous information. For information escalated applications, the MapReduce system has been broadly utilized as of late. In this paper, what's more, we apply the strength control sifting technique to viably prune non-horizon focuses ahead of time. We next segment information in light of the districts separated by the quad tree and figure hopeful horizon focuses for each segment utilizing MapReduce. At long last, we propose a productive technique for preparing multi-horizon questions with MapReduce with no change of the Hadoop internals. Through different analyses, we demonstrate that our approach beats past investigations by requests of extent.

Keywords — Hadoop, MCC, MapReduce, MDFS.

1. INTRODUCTION

multi-query processing in MapReduce (MR) has attracted great attention since the data size and the number of queries in many applications increase dramatically. Online retailers that serve a large number of customers, e.g., Amazon and Ebay, try to provide personalized daily report services. The report of each user is made based on his/her preferences (or purchasing patterns) e.g., ranges of prices, ranges of review scores, categories, etc. Skyline queries are useful to support this kind of services because they retrieve only the interesting items satisfying various user preferences. As the number of users rapidly grows, the importance of efficiently processing a large number of personalized skyline queries will also increase. There have been some studies on processing skyline queries in the MR framework. Most of them, however, focused on processing of a single skyline query. Recently, some studies on multi-query optimization in MR have been made. They proposed techniques that enabled systems to reduce redundancies between multiple queries. However, these studies show limitations to process skyline queries efficiently and also require modifications of the Hadoop internals. In this paper, we propose an efficient method to process multiple skyline queries with MR by reducing redundancies without modifying the Hadoop internals.

2. SYSTEM IMPLEMENTATION

We introduce two baseline methods for processing multiple skyline queries without modifying the Hadoop internals. We will compare the performance of the proposed method with these methods. This is because, as far as we are aware of, there is no previous work on the problem of multiple skyline queries on the MR framework.

Native approach: This approach processes multiple queries independently. A MR job is executed for each query, where the common input file is scanned for each job. In the map phase of each job, the common input file is scanned to generate a map output for the corresponding query. In the reduce phase of each job, the map output is distributed to the reducer and used for skyline calculation.

SH-scan approach: This approach provides sharing of input scan for multiple queries. In the map phase, the common input file is scanned once to generate map outputs for multiple queries. In the reduce phase, if the corresponding query of the map output is qi, we distribute that map output to the reducer of qi. Finally, each reducer calculates a skyline by using the given map output.

3. EXISTING SYSTEM

- MR-GPMRS consists of the partitioning andglobal skyline phases. The partitioning phase of MR-GPMRS divides the data space into grid partitions and prunes thepartitions that cannot contain any skyline point by utilizingthe dominance relationships between grid partitions. In theglobal skyline phase, in every unpruned partition P, thepoints which are located in other unpruned partitions andmay dominate a point in P are first collected and each pointin the partition P is compared with the collected points todetermine whether it is a global skyline point in parallel.
- To compute the skyline efficiently, an additional local skyline phase is involved between the partitioning and global skyline phases in MR-BNL, PPF-PGPS and SKY-MR. They compute the local skyline for each partition and use them to compute the skyline in the global skyline phase.

DISADVANTAGES OF EXISTING SYSTEM:

- Since the skyline algorithmsusing MapReduceincluding SKY-MRignore workload balancing of available machines, their performancesdegrade with increasing the number of machines.
- Computing a skyline is challenging today since we have to deal with big data

4. PROPOSED SYSTEM

- We proposethe MapReduce algorithm SKY-MR+ to compute skylineefficiently in this paper
- Our SKY-MR+ uses an adaptive quadtree building technique which splits each node judiciously depending on whether splitting the node is beneficial or not in terms of the estimated execution time. Among the skyline points in the region of the leaf node, we select the one such that the estimated number of checking dominance relationships between the pairs of points in the region is the smallest.
- To balance the workloads of available machines in the local and global skyline phases, we propose the workload balancing techniques to make the estimated execution times of all machines to be similar. Since our workload balancing problem is the same as the multiprocessor scheduling problem, which is NP-Hard, we use an effective approximation algorithm.
- We adapt the dominance-power filtering technique which maintains a set of dominating points that are expected to dominate many other points and we prune the points dominated by a dominating point.

ADVANTAGES OF PROPOSED SYSTEM:

- To demonstrate the efficiency and scalability of our SKYMR+, we compared SKY-MR+ to MR-GPMRS, MR-BNL,PPF-PGPS and SKY-MR by implementing them as well asconducting extensive performance study on Hadoop.
- Our experimental results confirm that SKY-MR+ is veryefficient and scalable compared to the other existing MapReducealgorithms including the state-of-the-art SKY-MR.
- For all data sets, since the dominance power filtering prunes non-skyline points in the local skyline phase resulting in the reduced overheads of computing the skylines and distributing the points via the network, it reduces the execution times of both phases.

5. LITURATURE SURVEY

Borzsonyi et al. [1] first consider how to efficiently obtain skyline objects in the database community, and propose two feasible algorithms BNL and DC. The BNL algorithm essentially compares each object in the database with all the others and returns the objects that are not dominated by any others. The DC algorithm divides the input objects into several groups that can fit in memory. The skyline objects in all groups are computed separately using a memory-based algorithm and then merged to produce the final result.

Chomicki et al. [2] propose the SFS algorithm which sorts the input objects according to a preference function and then returns the skyline objects in another pass over the sorted list.

Chan et al. [3] propose an effective approximate algorithm that is based on extending a Monte Carlo counting algorithm to fast return the skyline objects.

Huang et al. [4] present an efficient cell-dominance computation algorithm (i.e., CDCA) for processing arbitrary single subspace skyline query. The CDCA algorithm uses the regular grid index and prunes all the cells which are dominated by any other ones.

Li et al. [5] propose a system model that can support subspace skyline query in mobile distributed environment. This algorithm uses mapreduce and can obtain the meaningful subset of points from the full set of skyline points in any subspace.

Huang et al. [6] focus on supporting concurrent and unpredictable subspace skyline queries over data streams. To balance the query cost and update cost, the authors only maintain the full space skyline and then propose an efficient and scalable two-phase algorithm to process the skyline queries in different subspaces based on the full space skyline. Recently, several literatures study the extensions of

6. RELATED WORK

Procedure of Proposed Method The key idea of SH-adv approach is to calculate skylines in fragments prior to calculate skylines of queries. SH-adv approach consists of three steps below: Preprocessing. In this step, we make fragments from input queries by using Algorithm 1 and store two information; a set of fragments F and a mapping table FMAP to indicate that which fragment belongs to which queries.

Fragment-level processing. In this step, we produce skyline per fragment from the input file. In the map phase, we start with loading the set of fragments F. For each input tuple, we find the corresponding fragment whose data space includes that tuple. We then generate

the map output by using an input tuple as a value and an ID of the corresponding fragment as a key. In the reduce phase, we compute a skyline for each key, fragment ID, and produce output of the form (fragment ID, skylines). We call the result skylines of this step as fragment-skylines.

Query-level processing. In this step, we produce skyline per query by using the output of the previous step. In the map phase, we start with loading the mapping table FMAP. For each fragment in the input tuple of the form (fragment ID, skylines), we find queries which include that fragment by looking up the FMAP.

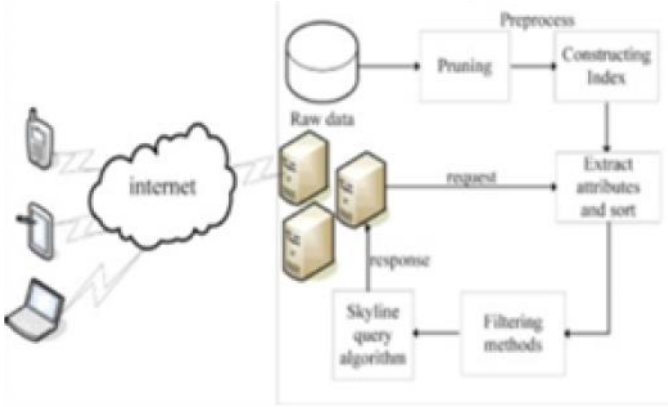

Fig: Architecture

CONCLUSION

In this paper, we proposed an efficient method to simultaneously process multiple skyline queries without modifications of the Hadoop internals. Our method was successful to process multiple skyline queries efficiently by considerably reducing redundancies in skyline query processing. The proposed method is composed of two consecutive MR jobs, but outperforms baseline approaches significantly by reducing the redundant generation of map outputs and skyline calculations as well as redundant input scans.

REFERENCE:

[1] J. Lee, S. won Hwang, Z. Nie, and J.-R. Wen, "Navigation system for product search," in ICDE, 2010.

[2] T. Lappas and D. Gunopulos, "Efficient confident search in large review corpora," in ECML/PKDD (2), 2010.

[3] G. Wang, J. Xin, L. Chen, and Y. Liu, "Energy-efficient reverse skyline query processing over wireless sensor networks," TKDE, vol. 24, no. 7, 2012.

[4] L. Zou, L. Chen, M. T. Ozsu, and D. Zhao, "Dynamic skyline ¨ queries in large graphs," in DASFAA, 2010.

[5] C. Kim and K. Shim, "Supporting set-valued joins in nosql using mapreduce," Information Systems, vol. 49, pp. 52–64, 2015.

[6] Y. Kim and K. Shim, "Efficient top-k algorithms for approximate substring matching," in SIGMOD, 2013, pp. 385–396.

[7] J. Dean and S. Ghemawat, "Mapreduce: Simplified data processing on large clusters," Communication of the ACM, vol. 51, no. 1, pp. 107–113, 2008.

[8] K. Mullesgaard, J. L. Pedersen, H. Lu, and Y. Zhou, "Efficient skyline computation in mapreduce," in EDBT, 2014, pp. 37–48.

[9] B. Zhang, S. Zhou, and J. Guan, "Adapting skyline computation to the mapreduce framework: Algorithms and experiments," in DASFAA, 2011, pp. 403–414.

[10] J. Zhang, X. Jiang, W. S. Ku, and X. Qin, "Efficient parallel skyline evaluation using mapreduce," IEEE Trans. Parallel Distrib. Syst., vol. 27, no. 7, pp. 1996–2009, 2016.

[11] Y. Park, J.-K. Min, and K. Shim, "Parallel computation of skyline and reverse skyline queries using mapreduce," VLDB, vol. 6, no. 14, pp. 2002–2013, 2013.

[12] R. L. Graham, "Bounds on multiprocessing timing anomalies," SIAM journal on Applied Mathematics, vol. 17, no. 2, 1969.

[13] Y. Park, J.-K. Min, and K. Shim, "Processing of probabilistic skyline queries using mapreduce," VLDB, vol. 8, no. 12, 2015.

[14] "Apache hadoop," http://hadoop.apache.org. [15] J. Chomicki, P. Godfrey, J. Gryz, and D. Liang, "Skyline with presorting," in ICDE, 2003, pp. 717–719.

[16] D. Kossmann, F. Ramsak, and S. Rost, "Shooting stars in the sky: An online algorithm for skyline queries," in VLDB, 2002.

[17] J. Lee and S.-w. Hwang, "Scalable skyline computation using a balanced pivot selection technique," Information Systems, vol. 39, pp. 1–21, 2014.

About author:

B.KAVITHA is presently working as Lecturer, Dept. of Computer Science, Sri Durga Malleswara Siddhartha Mahila kalasala,Vijayawada, A.P., India. She has ten years of experience in teaching field, her area of interests are Hadoop.