January – February 2020
ISSN: 0193-4120 Page No. 13388 - 13394

Skyline computation with Post Filter on Hadoop framework with Multiple Reducers

P. Venkateswara RAO[1], Dr. Mohammed Ali Hussain[2]

[1]Research Scholar, [2]Professor

[1,2]Department of computer Science and Engineering,KoneruLakshmaiah Education Foundation, Vaddeswaram, AP, India.

[1]pvrao.pd@gmail.com, [2]dralihussain@kluniversity.in

Article Info
Volume 82
Page Number: 13388 - 13394
Publication Issue:
January-February 2020

Article History
Article Received*: 18 May 2019*
Revised*: 14 July 2019*
Accepted*: 22 December 2019*
Publication*: 24 February 2020*

Abstract

Today's world generating huge data that shows exponential growth of data space along with multi-dimensional complexity, with this data query processing becomes difficult. Till now we are using traditional skyline to process all types of data sets. Skyline process demands more computational memory and efficiency for their traditional models. Skyline query processing suffering with the major problems due to increasing in time and space complexity over imbalanced datasets. There is chance of increasing duplicate data sets is another limitation for using traditional skyline processing models to handle spatial uncertain data. Parallel Skyline computation is one solution to handle large spatial data. Parallel skyline computational models are already available, but they also suffering with producing duplicate data sets. By keeping this in mind another solution is proposed that is filtering the data to identify and remove duplicate sets. This idea works and improves query processing speed by eliminating sparsity or empty patterns. In this paper we used k-nearest neighbor approach to eliminate empty pattern or sparsity. If we reduce empty pattern and sparsity it helps to reduce query processing time.

Keywords:*Skyline processing, parallel processing, spatial data, Hadoop, GPMRS, and MapReduced frame work.*

I. INTRODUCTION

In Internet there is a rapid growth of Data repositories by collecting data from different sites. Different sites provide data in different formats. There is a centralized database that contains heterogeneous data, which collects from different sources. Computing heterogeneous data with traditional skyline faces computational issues with respect to size and time. In heterogeneous database it is necessary to find the relation between essential spatial frequent and infrequent patterns.

To find hidden knowledge for available database is required to improve decision making. In Special Data mining it is required to gain frequent skyline points. Evolutionary approaches can't handle high dimensional data, due to time and memory limitations. Divide and conquer approach can be used to resolve this problem. For dived and conquer approach-based MapReduce model can be used on large spatial datasets with limited dimensions. Here Data is processed parallel after splitting it as sub problems. Still there is a chance of failure for divide and conquer method in high dimensional database to get the majority and minority spatial patterns. Known skyline parallel processing models consumes significant amount of time for medium to high database scans. Methods to improve the time complexity of pattern mining process are influenced by number of candidates sets it generates.

Published by: The Mattingley Publishing Co., Inc.

TEST Engineering & Management

January – February 2020
ISSN: 0193-4120 Page No. 13388 - 13394

Consider two points X and Y in n-dimensional space. Where X is dominates Y, if Y is not best than X with all dimensions and X is best at least with one dimension compared to Y. Let us consider skyline dataset 's' is a subset of 'S'. Where all the points in dataset 's' are not dominate by any other points in 'S'. Skyline operator is used to monitor and analyze uncertain stream of data.

For example there are some apartments near to bus terminal, they are consider as skyline points , if any of the apartment are not having lower rent and even though their distance is same, or if no apartment is near to the bus terminal having less rent. Now we have to analyze the apartment near to the bus terminal and having less rent. To compute this, we can go with skyline, but it is complex when size of the dataset is increased. To overcome this, we can go with parallel skyline computation. Parallel skyline computation divides the dataset into subsets before processing. Initially Local skyline points are derived by computing subsets. Then global skyline points are derived from local skyline points. Parallel processing computation divides data into subsets based on the machine's availability. There is no difference in the structure for the original and subset datasets. To process the skyline on local machine R-Tree indexing structure is used.

We are using Hadoop for implementing parallel processing in skyline. Traditionally there is a coordinator act as a central server. Each and every coordinator control N server. Because of large memory requirements, to compute high dimensional and cardinal data coordinator distribute the processing tasks to their N servers. Coordinator collects the Local skyline objects generated by individual servers. The coordinator sequentially processing local skyline objects to generate global skyline. High computational servers with memory and storage required for sequential skyline computation.

To process High dimensional datasets, we required parallel multiple cluster nodes. This environment is available in Hadoop with Map-reduce framework.

Apache Hadoop is best suitable open-source distributed framework to work with large amount of data. This Map-Reduce framework automatically divides data into segments. Cluster nodes will take care to execute those segments and to handle node failures. Hadoop requires the following components they are Hadoop File System (HDFS), Map-Reduce. HDFS maintains data in the form of blocks; each block size is 64MB. Hadoop components consumed resources are efficiently provided by Hadoop Core. An Efficient Map-Reduce framework improves the performance of the system with independent of Hardware and distributed environment [1]. To get effective interface, Map-Reduce on multiple cluster machines are used for skyline computation.

II. RELATED WORK

There is an efficient mining techniques developed by Y.Huang, J.Pei and H.Xiong [2] for skyline collocation rare events on special datasets. They proposed two measures; they are maxPR (maximal participation ratio) and pruneMAX (pruning Maximum). maxPR collects spatial co-location patterns with rare features, and pruneMax exploiting weak monotonicity. Easily to determine rare collocation patterns in skyline both maxPR and pruneMAX measures helps. These techniques are applicable only on Boolean spatial features. Real world features are categorical and continuous in nature. F K Deeb and L Niepel [3] are used KD-tree for efficiently finding spatial co-location patterns. This method uses group of instances to find spatial co-location patterns.

There is another method for mining collocation patterns by using an ordered neighbourhood method is given by Y.Cheng, et.al [4]. They used a special technique called Maximal clique enumeration over spatial data. It is used to filter the spatial co-location objects and patterns based on Order star

Published by: The Mattingley Publishing Co., Inc.

January – February 2020
ISSN: 0193-4120 Page No. 13388 - 13394

Neighborhood method. Zang et.al [5] addressed the problem of uncertain data with top-k skyline computational model. They used two procedures for data filtering and randomization process. To compute skyline probabilities based on grid pruning method is proposed by Pei et al. [6].They identified problem with parametric equations and that is replaced with multivariate probability density function.

Foto N. Afrati et.al [7] proposed Map-Reduce frame work to process skyline query with parallel processing. In this model to optimize query processing they minimized the load balancing steps. For high dimensional datasets it is not suitable. This is applicable for limited datasets. Next AkriviVlachou et.al also proposed skyline computation with Map-Reduce framework [8]. This is also applicable for Limited size datasets. Map-Reduce having two phases, they are Mapper and Reducer phase. In this model the Mapper is used to partition data and tried to reduce noisy objects using Mapper function. Angle based partitioning method is used to process partitioned objects. With reducer function the data objects further it will be processed and aggregates. Here we will get Local skyline objects from angle-based partitioning. Then reducer produces Global skyline objects after merging intermediate results.

Grid-based technique is introduced by F He, X Deng, J Fang [9] , this is a new technique for spatial co-location mining. This Grid-based technique is a new technique for spatial co-location mining. For Efficient co-location mining computation, they are using parallel programming in skyline. In this method the entire spatial space is divided as grids. Here the grid is formed with a spatial space that split as small cells. The length of the cell will play major role here, because if cell is too small then it increases computational time. Mullesgaard [10] proposed Map-reduce based skyline with single reducer for processing grid partitioning. Bit string representation is used to represent data space that is split as grid.

The skyline processing using multi-core architecture is proposed by Hyeonseung et.al, [11]. They used PSkyline and SSkyline as new measures for skyline computational. BBS, SFS are used for PSkyline measure. Due to huge set of candidate sets and null patterns, proposed method is not suitable with GRID based skyline objects. Park proposed Map-Reduce based Skyline called SKY-MR [12]. Before start MapReduce process Sky-quad tree is built for dataset for locating the dominated sample regions. The data space is repeatedly breaking down as sub-regions with Sky-quad tree. There is a chance of computational over head with continuously updating of sky-quad tree due to large scale of data sets.

III. PROPOSED MODEL

To process high dimensional spatial data sets in skyline we use multiple reducer Haoopframework with filtered grid partition. This filter-based grid partitioning method will reduce the processing time. Parallel processing feature solves the problems with grid and angular based skyline query processing.

The following features used to solve problems identified with angular and grid partition processing. Each cluster node in parallel skyline computation processes the query and sends to coordinator for filtering. Scalability can be handling by spreading participating nodes in sky-line processing, this leads to workload distribution. In Map-reduce for candidate filtering intermediate result are used. Intermediate results are local data skyline processing result by each node. To ensure each node to contribute global skyline point, Local skyline points are uniformly distributed to the servers. Individual mapper and reducer nodes use independent skyline algorithms for processing locally stored data points.

Hadoop is used to implement proposed framework. First phase of our proposed framework partitions the data into multiple sub sets for bit-string representation. To obtained bit-string representation

Published by: The Mattingley Publishing Co., Inc.

January – February 2020
ISSN: 0193-4120 Page No. 13388 - 13394

for sub-string Multiple Map-reducer jobs are used. To give general view and pruning out data partition of the data without skyline tuple is possible with bit-string. With the help of FGPMRS the original tuples are distributed among multiple mappers. The FGPMRS performs parallel computation on local skyline. The reducer phase collects all local skyline points from multiple mappers. The FGPMRS model performs the entire process in three stages: 1. preparing bit-string representation for pre-processing phase. 2. Using KNN-approach for filtering Local skyline. 3. Using KNN-approach for filtering Global Skyline. In pre-processing phase n x n grid is used to partition the data before pre-processing phase. The nd is the grid cell dominance. In d-dimensional space data is split into n parts for each of the dimension. In a Map-phase a cell is marked as "1" indicates that all non-empty partitions dominating that cell, for remaining cells which are marked as "0". The reducer merges all local skyline points to produce global skyline.

To reduce execution process time in entire process for large datasets, apply F-GPMRS, that method removes candidate sets before skyline computation.

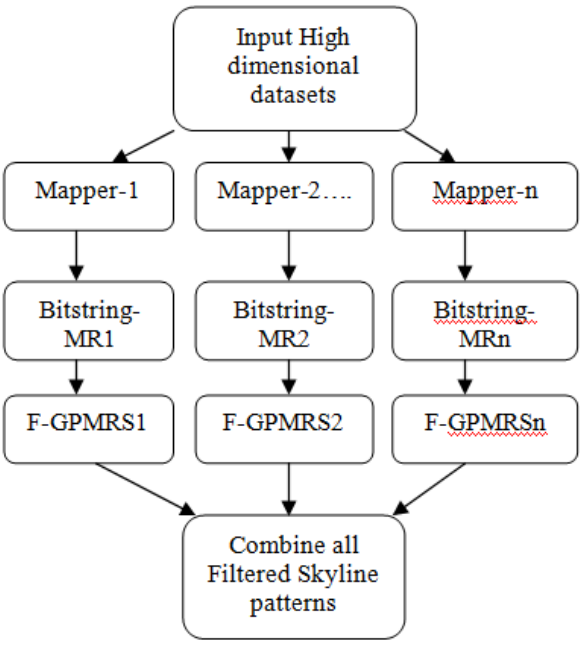

Figure 1: proposed FGPMRS Framework

This proposed model is an extension of GPMRS [10] using Hadoop framework.

Step1: Bit String representation:

Here skyline computational importance is given to non-empty tuples set R. Bit-string is represented from n x n partition.

BitStringR[i] = 1, if pi != ϕ

= 0, otherwise --- (1)

MapReduce Using equation (1) for bit-string computation is performed in mapper. Mapper takes each sub-set that is disjoint from another sub-set as input.

Step2: Applying GPMRS: This is applied with multiple mapper and reducers. Here Mapper derives Local skyline points and global skyline derived by reducers.

Step3: Applying Filtering method

To derived global skyline points from local skyline points, apply k-nearest neighbour algorithm for deriving nearest local skyline objects.

The O_i is independent partition group (IPG) object for each local skyline.

Do

For each object $O_i$is in IPG [ ]

Do

For each object O_j in IPG [ ] //where $i \neq j$

Compute distance N_m^k using Manhattan distance.

// Neighbour k-points in each severity level

N_m^k [ ] = Found Top K - nearest objects ;

Assign a class to p′ based on majority vote:

c′ = argmaxy∑(xi, ci) belonging to S, I(y=ci)

Done

Published by: The Mattingley Publishing Co., Inc.

January – February 2020
ISSN: 0193-4120 Page No. 13388 - 13394

Done

Step4: Performing Global skyline computation

Step5: Multiple reducer results are combined to merge Global Skyline computational results.

IV. EXPERIMENTAL SETUP

This proposed model along with other models is tested on Intel core i5 processor with 2.4 GHz speed and 8 GB internal Memory. To develop proposed system we used Jama, Jfreechart, scala, spark and third-party libraries.

Sample Anti-correlated datasets:

4519 3684 120 4604

4782 3613 2596 3055

607 3360 3675 3600

4548 2985 3928 2368

1201 1156 4643 3653

1304 2841 2997 4063

3922 4779 2965 3249

920 4078 2022 3706

2800 3675 4301 2523

1424 4414 4782 1348

2814 2266 2433 3656

1456 3888 4092 2439

3355 4670 1196 2688

3325 4361 85 4268

4345 2875 3120 191

3768 1547 2717 3715

3427 2765 2787 3868

4643 2585 3417 4076

2495 1483 3491 3036

4003 4841 752 2553

3458 4086 2712 383

3898 1470 822 4467

2542 4777 4200 3732

4715 2385 3211 1983

4953 1223 2381 3556

284 4926 3185 3086

101 3302 4947 2301

713 4419 4601 3233

1229 2303 4730 4299

4687 3070 4673 4815

3385 2588 4286 517

3643 1542 3501 2580

3332 2087 4861 1904

4320 2751 4161 2511

4039 4039 3166 4974

1751 3062 1184 4622

4720 2169 27 4323

1805 3489 3553 2072

1950 4004 4120 4350

1620 4801 4909 4082

4455 3941 4444 1087

We use spatial synthetic data with huge number of instances and dimensions.

Published by: The Mattingley Publishing Co., Inc.

January – February 2020
ISSN: 0193-4120 Page No. 13388 - 13394

Figure 2: Details of proposed model computation on 100000 skyline objects.

Table 1 compare the efficiency of three Algorithms with respect to runtime, dimensions, and average storage space in bytes. From this table we can come to a conclusion that FGPMRS consumes less storage space. These operations performed on 100000 records.

Model	Average Runtime (Sec)	Dimensions	Average Storage (bytes)
Angle based Skyline	63.64	5	20243
GPMRS	34.12	5	15293
FGPMRS	26.23	5	5932

Table1: Comparison of proposed model with the existing models with respect to Average runtime, dimensions and average storage are concerned.

Comparison of proposed models with existing models with respect to number of reducers, dimensions and average filtered pattern are shown in table2.

Model	Reducers	dimensions	Avg. Filtered Patterns
AngleSkyline	5	4	49
GPMRS	5	4	37
FGPMRS	5	4	10

Table 2: Comparison of proposed model with the existing models with respect to number of Reducers, dimensions and empty patterns.

From this table we can come to conclusion that our FGPMRS has fewer null patterns compared to the existing models.

V. CONCLUSION

We studied different skyline algorithms and identified increasing in computational memory and efficiency requirements due to increasing in growth of multidimensional objects. Using parallel processing feature we overcome the problems in traditional angular and grid-based skyline computational algorithms. We introduced a novel filtered based grid partitioning on high dimensional

Published by: The Mattingley Publishing Co., Inc.

January – February 2020
ISSN: 0193-4120 Page No. 13388 - 13394

spatial data sets is called FGPMRS. Proposed model is described with three phases as shown in figure 1. This is an extension of GPMRS model. FGPMRS removes more candidate sets before skyline computation that leads to reducing processing time. When we have taken 1 lack records with dimensions 6 then FGPMRS shows average running time of 27 seconds, and average storage is 6386 bytes this is very less compared to another algorithm. Scope of this paper is further to decrease performance time.

REFERENCES

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

[2] Yang Cheng; Zhang Tianjun; Lu Junli," A Maximal Clique Enumeration Based on Ordered Star Neighbourhood for Co-location patterns " , in 2013 5th International Conference on Intelligent Human-Machine Systems and Cybernetics, 2013, pp. 164 - 167.

[3] Yoon Jae Park, Jun-ki Min and Kyuseok Shim, "Efficient Processing of Skyline Query Using MapReduce", IEE Transactions on Knowledge and Data Engineering, 2017.

[4] Mullesgaard, Kasper; Pedersen, JensLaurits; Lu, Hua; Zhou, Yongluan. Efficient Skyline Computation in MapReduce, Proc. 17th International Conference on Extending Database Technology (EDBT), Athens, Greece, March 24-28, 2014.EDBT, 2014. pp. 37-48.

[5] Fei He1,2, Xuemin Deng3, Jinyun Fang1," A fast approach for spatial co-location pattern mining", 2013 IEEE International Geoscience and Remote Sensing Symposium-IGARSS,2013, pp.3654 – 3657.

[6] Yan Huang•JianPei•HuiXiong, "Mining Co-Location Patterns with Rare Events from Spatial Data Sets ", Geo informatics, 2006, pp. 239–260.

[7] Foto N. Afrati, ParaschosKoutris, Dan Suciu, Jeffrey D. Ullman, "Parallel skyline queries", Proceedings of the 15th International Conference on Database Theory, March 2012, pp. 26-29.

[8] Parallel Skyline Computation on Multicore Architectures, Sungwoo Park, Taekyung Kim, Jonghyun Park, Jinha Kim, HyeonseungIm, Department of Computer Science and Engineering, Pohang University of Science and Technology Gyeongbuk, Republic of Korea, pp. 790-784.

[9] Katja Hose, AkriviVlachou, "A survey of skyline processes in highly distributed environments", The VLDB Journal (2012), PP: 359–384.

[10] Jian Pei, Bin Jiang, Xuemin Lin, Yidong Yuan, "Probabilistic skylines on uncertain data", Proceeding VLDB '07 Proceedings of the 33rd international conference on Very large databases, PP: 15-26.

[11] Ying Zhang, Wenjie Zhang, Xuemin Lin, Bin Jiang, Jian Pei, Ranking uncertain sky: The probabilistic top-k skyline operator, Information Systems 36(2011), PP: 898-915

[12] Fadi K. Deeb; LudovicNiepel," A methodology for discovering spatial co-location patterns", 2008 IEEE/ACS International Conference on Computer Systems and Applications, 2008, PP: 134 – 141.

[13] P.VenkateswaraRao, Dr.Mohd Ali Hussain, "Mashup service implementation on multi-cloud Environment using Map Reduction Approach", Journal of Advanced Research in Dynamical and Control Systems, 2017, PP: 758-767.

Published by: The Mattingley Publishing Co., Inc.