Journal of Theoretical and Applied Information Technology
20th July 2015. Vol.77. No.2

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

EFFICIENT PROBABILISTIC APPROACH TO COMPUTE SKYLINE SET IN DISTRIBUTED ENVIRONMENT

[1] T.VIJAYA SARADHI, [2] K.SUBHRAHMANYAM [3] L.JAGAJEEVAN RAO,

[1]Asst Prof., Department of Computer Science and Engineering, K.L University, Vaddeswaram
[2] Prof., Department of Computer Science and Engineering, K.L University, Vaddeswaram
[3]Asst. Prof., Department of Computer Science and Engineering, K.L University, Vaddeswaram

E-mail: [1]saradhi1440@kluniversity.in, [2]smkodukula@ kluniversity.in, [3]jeevan.lingampalli@gmail.com

ABSTRACT

The Uncertainty in a distributed environment is a major challenge due to communication delay, limitations of measuring data and network latency. Due to this uncertainty, a considerable amount of work has been dedicated to handle inconsistent data and resolving simple queries on inconsistent data, but the best way to handle complex uncertain data continues to be an open problem. Probabilistic skyline is one of the enhanced solutions to handle uncertainty in a distributed environment. The probabilistic based approach has been extensively used in data mining, multiple decision-making criteria and business planning. The skyline of multidimensional data includes those points which are not dominated by any other point in all n-dimensions. Probabilistic computation of the skyline considers each point to be represented by a n-dimensional data point. Mostly, the distributed skyline queries are designed to find a set of non-dominated data objects in a multi-dimensional data. Most of the conventional work has assumed a distributed or centralized configuration, in this paper, an efficient probabilistic computation of skyline queries on large distributed data was proposed. The experimental results show that the efficiency of the probabilistic based approach and its limitations in terms of time and accuracy are concerned.

Keywords: *Skyline Computation Distributed Environment, Probabilistic Skyline Computation.*

1. INTRODUCTION

Actually, the distributed skyline query as an essential feature of big-data mining. For example, consider the resource selection in distributed environments like cloud where users may wish to select optimal resources or services to meet their Quality of Service specifications. The geographical positioned cloud servers will provide a large number of related services with different features such as stability, response time, cost, reliability, accuracy, and elasticity and different prices [1]. Thus, modeling the solutions as uncertain data and assessing them with skyline queries is an important task in large databases. Skyline queries, in the context of distributed databases, were initially implemented in [2] due to their applicability in multi criteria decision making approach, without the problem of user defined scoring features.

The original concept of skylines for n-dimensional feature spaces signifies the concept of dominance as follows: Point pt1 dominates the point pt2 if point pt1 is smaller in at least one and less than or equal to in all other n-dimensions than point pt2. The skyline operator f_{sky} retrieves all points of a database which are not dominated by any other database point. For a database DB consisting of n-dimensional points with pt1, pt2∈ DB can be expressed as:

f_{sky}(DB) = DB: {pt1|∃p ∈ DB (∀i ∈ {1,...,n} : pt1(i) ≤ pt2(i))}

An example for the skyline operator for hotel dataset retrieves the most relevant hotels where each hotel is a two-dimensional feature object indicating the distance to the beach as y-axis and the price of the hotel as x-axis. However, this “Hotel example” is generally used for describing the skyline operator; it implies that both, the price and the time to the beach can be precisely specified. In most cases, a hotel will provide different rooms having different rates. Additionally, the time to the beach relies upon whether you take a car, walk, or the transportation system. Thus, a hotel can be designed more appropriately, if we illustrate each dimension by probability density function (pdf) [4].

Journal of Theoretical and Applied Information Technology
20th July 2015. Vol.77. No.2

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

As shown in Fig. 1, where points are scattered over two cloud servers. The set of local skyline points in each server is shown by using a two different line style, whereas filled points are the global skyline.

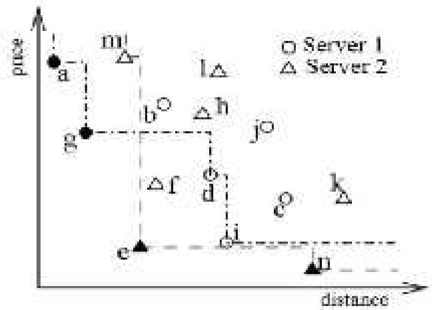

Fig 1: Scattered Points Over Two Cloud Servers

As a second illustration, consider a multimedia dataset where pictures are three dimensional objects, with each pixel being the mean value of red, green and blue in it. User p1 interested images that are less red, but more blue and green, whereas user p2 is interested in images that are less green but higher red and blue. Evidently, p1 generates a skyline query Qu1 with preferences <min,max,max>, whereas p2 asks for <max,min,max> in his query Qu2.

Skyline: Let S1 be the set of all points in the distributed data, D denotes the total dimensions. A object p1 ∈ S1 is said to dominate another object q1 ∈ S1, denoted as $p1 \prec q1$, if (1) On every dimension dim ∈ D, p1 [j] ≤ q1 [j]; $\forall j$ and (2) On at least one dimension d ∈ D, p1 [j] < q1[j] $\forall j$.

Each non empty subset S1 of D (S1⊆ D) is known as a subspace of D. Each data vector space D is also known as the full space of the dataset S1. A point p1 ∈ S2 is said to dominate another point q1 ∈ S2, denoted as $p1 \prec q1$, if (1) On every dimension dim ∈ D, p1 [j] < q1[j] $\forall j$ and (2)At least one object d ∈ D, p1 [j] < q1[j] $\forall j$. The subspace S2⊆ D is a set of skyline sub points with n-dimensions which is not dominated by any other object in the subset. Consider, for instance the data depicted in Fig 2. Skyline objects are Spts = {k,i,j}, while for the subspace S2 with the skyline points on complete skyline objects is represented as S2= {i, m}.

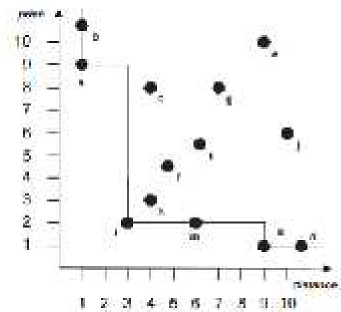

Fig 2: Skyline Points View Of Hotel Data

2. LITERUTARE SERVAEY

2.1Skyline Computation In Distributed Environment

A framework, called SkyPlan [2] has been implemented that finds the dependencies between the skyline queries in a directed graph with cost-aware plans. One of the possible methods to deal with geographical data has been {examined} using an approach called PadDSkyline [3]. In [9], the authors have suggested a method called as Search Space Partitioning, which improve capabilities of Balanced Tree Overlay network [10] for indexing the data so that in structured p2p network, the peers will be utilized accurate subspace data to compute the relevant skyline points can be located effortlessly. Kossmann, Ramsak and Rost [15] implemented a Nearest Neighbor technique to process skyline queries repeatedly. It first transmits out a nearest neighbor result on the dataset located in an R^*-tree, and then inserts the new point into the skyline detection. The new NN point also decides a region which only includes points dominated by NN and thus can be pruned. The remaining space is partitioned into two regions based on the new NN point, and both are addedinto a priority list. The progressive skyline computation [5] and other algorithms [3] have also been implemented for query load balancing in terms of data processing and query processing time. The emergence of multi core processors is making an intense impact on software design. The concept of incomparability for skyline computation has also been searched. A distributed skyline query can be extracted by assessing distributed skyline queries on different cloud servers. Other aspects of parallel skyline computation like constrained skyline queries, rank-aware queries and progressive skyline computation in peer to peer networks have been proposed in [4-6,11-13].In differ to uncertain data, existentially uncertain data [10] are precisely specified, but have a possibility of existence to monitor offshore

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

traffic, resource planning and especially regarding scheduling. Due to restricted image quality resolution, it might be impossible to constantly decide whether a pixel mixture represents an object or not. In this case, the proposed approach assigns an estimated probability value to each pixel region, which signifies the confidence that the object actuallyexists [14].

3. METHODOLOGY

3.1Traditional Distributed Skyline Approach for Uncertain Data

In this approach, a general workflow for processing skyline queries on the distributed data was implemented. If the users wish to retrieve the high dimensional skyline set, this system can be easily applied to any attributes of size d, that is, simply by evaluating the dominant conditions between attributes only on the userchosen k dimensions. Fig. 1 illustrates the general workflow for answering skyline queries over noisy or uncertain dataset[1].To process the probabilistic skyline that retrieves the points whose probabilities of being skyline is bigger than the user defined threshold p, the Top-up and Bottom-down algorithms are proposed in [16] to enhance the query. Besides, all skyline query issues are extensively handled in [17]. Moreover, Böhm et al. [18] survey the continuous case of the p-skyline query where each point is modeled as a distribution. To optimize the query, all the points can be indexed with the Gaussian based tree in the space. Zhang et al. [19] extend the p-skyline operator to data-streams with a sliding window pattern. Li et al. [20] handle the parallel skyline queries over uncertain data-streams with partitioning and grid index. Besides, Lian and Chen [21] first present the concept of probabilistic reverse skyline and proposed techniques for filtering the query including the monochromatic.

At the beginning of the distributed skyline detection for uncertain data, each site has calculated its local site skyline value using a central skyline method and arranges the skyline elements in decreasing order of their p-skyline values. Simultaneously, the central representative server H assigns an empty-set Si which calculates its local skyline value using a centralized skyline method and arranges the skyline elements in decreasing order of their skyline probability values. Then, the remaining part is executed in iterations until all the local skyline sets are null or the local skyline probability of the tuple within D_i is less than the user given threshold t. Each iteration in this workflow has four phases [1]:

3.2 K-Site Parallel Processing

In the first phase, each local-site S_i divides D into k-partitions with k-data representative objects. Each partition finds the highest probability value as its representative object. Partition with highest representative point is selected as base partition.

Algorithm:

Input:

D((v1,v2…vn),prob)

D_{s_i} : Denotes ith site data.

P(s_{ij}): represents i^{th} site j^{th} partition.

r(P(s_{ij})): denotes representative point in the partition.

Output: K-representative points.

Procedure:

For each site s_i

do

P(s_{ij}):=Random(k, D_{s_i}) // randomly partition the site data into k partitions.

Done

For each partition j in s_i

Do

Choose initial representative point as

r = highest Dominant point in P(s_{ij})

₵ :=Max(| $r - v[i][j]$ |)/n where n denotes partition size

If(r> ₵)

Then

Select r as partition representative point.

r(P(s_{ij})):=r;

end if

done

After the execution of the first step, each site transfers its representative tuple to the server based on the outcome of the server broadcast phase. In the second phase, central coordinator H combines all the representative skylines tuples from distinct sites S_i into a sub dataset D_0, and gets the skyline set $sky(D_0)$ of D_0 using a centralized skyline method.

In the third phase, Centralized server H chooses a tuple from $sky(D_0)$ with the largest p-skyline value and sends it to the remaining local sites in

Journal of Theoretical and Applied Information Technology
20th July 2015. Vol.77. No.2

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

order to get its g-skyline probability.Finallyin the fourth phase, the selected tuple from the centralized sever H is delivered to the local sites, a local filtering technique is executed to prune all those un-qualified tuples from $sky(D_0)$.

According to [1], for a tuple s with site D_j the global skyline probability denoted as $g-sky_p(s)$ is the upper bound of the local p-skyline $l-sky_p(s_j, D_i)$ multiplies the upper-bound of its local p-skyline values to the remaining uncertain databases D_{oth}.

$$g-sky_p(s) = \prod_{oth}^{m} l-sky_p(s\ , D_{oth})$$

$$g-sky_p(s) = l-sky_p(s\ , D_j) * \prod_{oth=1, oth \neq j}^{m} l-sky_p(s\ , D_{oth})$$

$$g-sky_p(s) \leq l-sky_p(s\ , D_j) * \prod_{oth=1, oth \neq j, t \in D_{oth}, t \prec s}^{m} l-sky_p(s\ , D_{oth})$$

$$g-sky_p(s) \leq l-sky_p(s\ , D_j) * \prod_{oth=1, oth \neq j, t \in D_{oth}, t \prec s}^{m} \{l-sky_p(t\ , D_{oth}) / p(t)\} * (1-p(t))$$

4. LIMITATIONS OR FUTURE DIRECTIONS:

1) As the dimension size increases the global skyline prediction time also increases.
2) Most of the distributed skyline queries depend on the user specified threshold.
3) As the data size or number of sites increases, the minimum bounding rectangle area also increases. So it is difficult to find the optimal skyline points within the bounded rectangular regions.
4) Since the data is distributed among multiple sites, this approach fails to find the outliers or inconsistent data points in the server phase.
5) As the volume of candidate sets at the data server increases, then there is a need to apply parallel skyline computation for subsets of data.

5. EXPERIMENTAL RESULTS

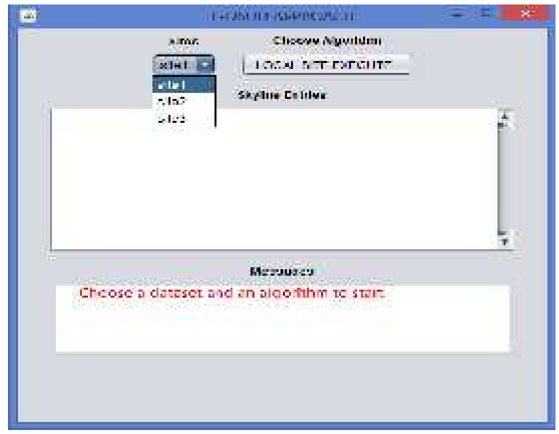

Fig 4: Home View Of Distributed Skyline Queries

Site one: Parallel computation results
Representative Point in Partition Point [x=277.0, y=256.0]
Representative Point in Partition Point [x=160.0, y=980.0]
Site one Skyline Entries:
Skyline entries 6
Skyline probability Point [x=429.0, y=172.0]
Skyline probability Point [x=523.0, y=110.0]
Skyline probability Point [x=812.0, y=106.0]
Skyline probability Point [x=131.0, y=921.0]
Skyline probability Point [x=88.0, y=3383.0]
Skyline probability Point [x=86.0, y=4541.0]

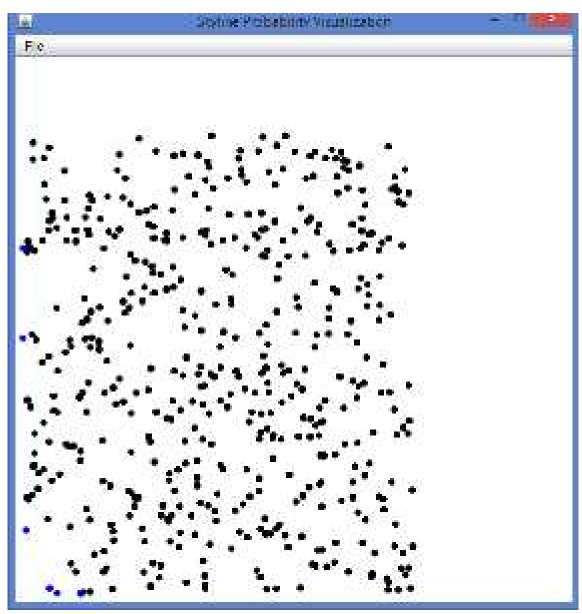

Fig 5: Site One: Probabilistic Skyline Queries Over Uncertain Data

Journal of Theoretical and Applied Information Technology
20th July 2015. Vol.77. No.2

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

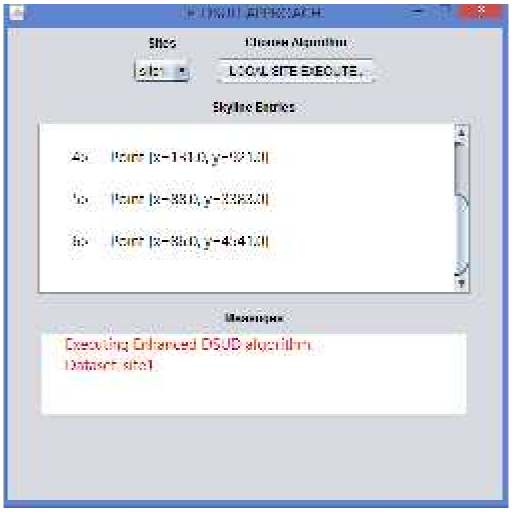

Fig 6: Site Two Execution Process

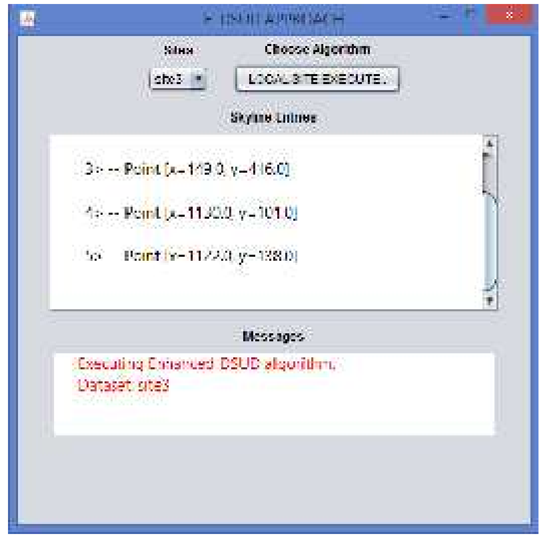

Fig 8: Site Three Local Executions

Site two skyline entries:
Skyline probability Point [x=198.0, y=280.0]
Skyline probability Point [x=382.0, y=116.0]
Skyline probability Point [x=175.0, y=631.0]
Skyline probability Point [x=86.0, y=1477.0]
Skyline probability Point [x=83.0, y=1999.0]

Site Three: Skyline entries
Skyline probability Point [x=254.0, y=150.0]
Skyline probability Point [x=100.0, y=430.0]
Skyline probability Point [x=149.0, y=416.0]
Skyline probability Point [x=1130.0, y=101.0]
Skyline probability Point [x=1122.0, y=138.0]

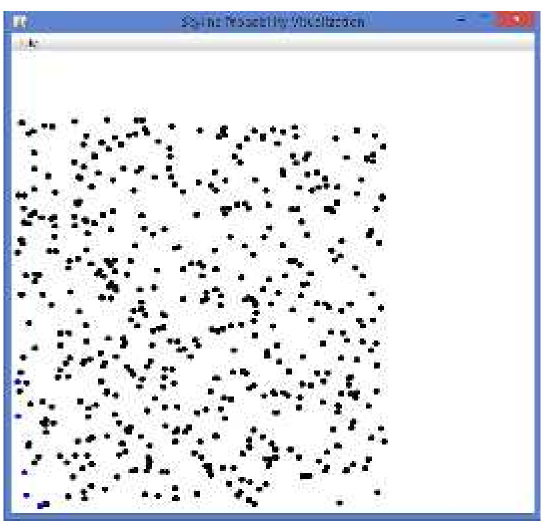
Fig 7: Site Two: Probabilistic Skyline Queries Over Uncertain Data

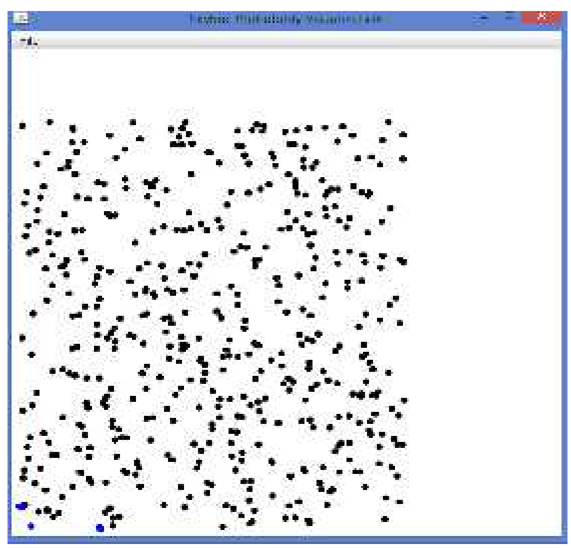
Fig 9: Site Three: Probabilistic Skyline Queries Over Uncertain Data

Journal of Theoretical and Applied Information Technology
20th July 2015. Vol.77. No.2

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

To Server Phase Results:

Site one:

523	110	3092.329	2515.536	0.747158	0.93423
131	921	2419.973	4228.024	0.233993	0.71325
86	4541	3496.407	1957.491	0.424145	0.125272
429	172	1912.59	711.534	0.283937	0.093182
812	106	639.613	1401.586	0.095918	0.469259
88	3383	3496.773	2887.364	0.000409	0.951315

Site two:

198	280	1417.947	1450.187	0.238301	0.911583
175	631	2863.443	4575.009	0.085612	0.147053
86	1477	508.88	4703.742	0.609645	0.372606
83	1999	3975.778	3008.08	0.064914	0.561376
382	116	3926.338	3368.448	0.066735	0.970894

Site Three:

254	150	2281.946	2646.868	0.400324	0.29823
1130	101	2690.058	582.196	0.123705	0.594178
1122	138	2542.177	4095.59	0.223113	0.121492
100	430	1353.44	3209.447	0.881081	0.620406
149	416	2577.658	621.028	0.402142	0.020631

Fig 10: User Selected Threshold

Server Computation

Final Server H 86.0 4541.0 3496.407 1957.491 3496.407 0.609
Final Server H 1130.0 101.0 2690.058 582.196 2690.058 0.793

5.1 Performance Analysis

Table 1: Skyline Computation In Terms Of Number Of Sites And Memory Usage.

Number of Sites	p-skyline points	ServerComputation(ms)	Memoryusage(MB)
2	15	156	15
3	18	234	17
4	15	289	25
5	19	326	29
6	15	453	36
7	14	**524**	42

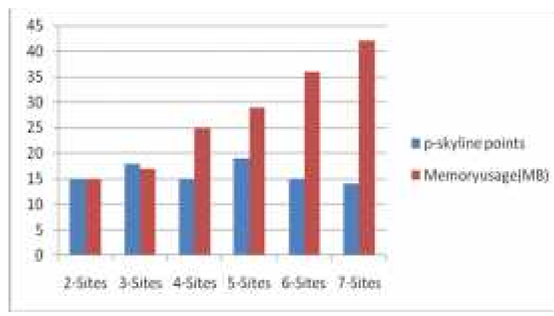

Fig 11: Skyline Points Vs Memory Usage

Sites	EDSUD	Proposed
Sites_2	**156**	**148**
Sites_3	**234**	**213**
Sites_4	**289**	**263**
Sites_5	**326**	**322**
Sites_6	**453**	**398**
Sites_7	**524**	**489**

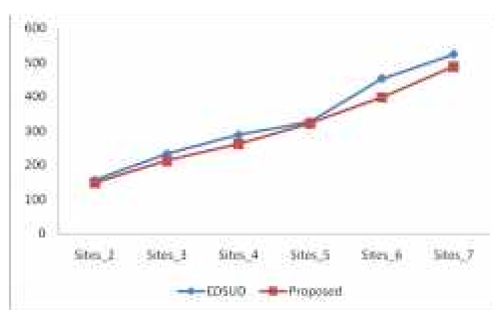

Fig 12: Server Computation Cost In Existing And Proposed Work

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

REFERENCES

[1] Xiaofeng Ding and Hai Jin, "Efficient and Progressive Algorithms for Distributed Skyline Queries over Uncertain Data", IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, VOL. 24, NO. 8, AUGUST 2012.

[2]. AkriviVlachou,Christos Doulkeridis, João B. Rocha-Junior, and KjetilNørvåg, Efficient Execution Plans for Distributed Skyline Query Processing, March 2011.

[3]. Bin Cui, Hua Lu, Lijiang Chen, Constrained Skyline Query Processing against Distributed Data Sites,IEEE TKDE, pp. 204-217, Vol. No 23 no 2, Feb 2011.

[4]. Seung-won Hwang, Jngwuk Lee, BskyTree: Scalable Skyline Computation Using A Balanced Pivot Selection, Lausann, Switzerland, 2010, pp. 195-206.

[5]. Ping Wu, Ying Feng, Caijie Zhang, DivyakantAgrawal,, Ben Y. Zhao, and Amr El Abbadi ,Parallelizing Skyline Queries for Scalable Distribution,LNCS 3896, pp. 112–130.

[6]. Lin Zhu, Jihong Guan, Shuigeng Zhou, Efficient Skyline Retrieval on Peer-to-Peer Networks, FGCN , 2007, pp. 309-314.

[7]. Jonghyun Park, HyeonseungIm, Sungwoo Park, Parallel skyline computation on multi-core architectures, ICDE, 2009, pp. 808-823.

[8]. AdanCosgaya-Lozano, Norbert Zeh,Andrew Rau-Chaplin, Parallel Computation of Skyline Queries, IEEE HPCS'07.

[9]. Beng chin Ooi,LizhenXu, Shiyuan Wang, Anthony Tung, Efficient Skyline Processing on Peer to Peer Networks, ICDE, 2007.

[10]. Lijiang Chen, Bin Cui, LinhaoXu,QuanqingXu,Hua Lu, Guojie Song, Efficient Skyline Computation in Structured Peer-to-Peer Systems, IEEE TKDE, Vol. 21, No. 7, July 2009.

[11]. M. Karnstedt, K. Hose, A. Koch, D. Zinn,K. Sattler,Processing rank-aware queries in P2P systems. DBISP2P, 205, pp. 238–249.

[12]. Q. Xu, L. Chen,B. Cui, H. Lu, Y. Dai, Y. Zhou, Parallel Distributed Processing of Constrained Skyline Queries by Filtering, IEEE. Int‟l Conf. Data Eng. (ICDE), 2008, pp. 546-555.

[13]. B. Cui, H. Lu, L. Chen, L. Xu, Q. Xu, iSky: Efficient and Progressive Skyline Computing in a Structured P2P Network, Proc..Int‟l Conf. Distributed Computing Systems (ICDCS), 2008, pp. 160-167.

[14] X. Dai, M. L. Yiu, N. Mamoulis, Y. Tao, and M. Vaitis. Probabilistic spatial queries on existentially uncertain data.In SSTD, 2005.

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

[16] J. Pei, B. Jiang, X. Lin, Y. Yuan, Probabilistic skylines on uncertain data, in: Proceedings of the International Conference on Very Large Data Bases (VLDB), 2007, pp. 15–26.

[17] M. Atallah, Y. Qi, Computing all skyline probabilities for uncertain data, in: Proceedings of the ACM Symposium on Principles of Database Systems (PODS), 2009, pp. 279–287.

[18] B. Christian, F. Frank, O. Annahita, Computing all skyline probabilities for uncertain data, in: Proceedings of the IEEE International Conference on Data Mining (CIKM), ACM, 2009.

[19] W. Zhang, X. Lin, Y. Zhang, W. Wang, J. Yu, Probabilistic skyline operator over sliding windows, in: Proceedings of the IEEE International Conference on Data Engineering (ICDE), 2009, pp. 1060–1071

[20] X. Li, Y. Wang, X. Li, Y. Wang, Parallelizing skyline queries over uncertain data streams with sliding window partitioning and grid index, Knowl. Inf. Syst., http://dx.doi.org/10.1007/s10115-013-0725-8.

[21] X. Lian, L. Chen, Monochromatic and bichromatic reverse skyline search over uncertain data, in: Proceedings of the ACM International Conference on Management of Data (SIGMOD), 2008, pp. 213–22621