Distributed Skyline Processing: a Trend in Database Research Still Going Strong

Katja Hose

Max-Planck Institute for Informatics
Saarbrücken, Germany

Akrivi Vlachou

Norwegian University of Science and Technology, Trondheim, Norway

NTNU
Norwegian University of
Science and Technology

Outline

- ***Skyline processing in unstructured P2P***
  - ***Existing approaches for skyline queries***
  - Skyline variants
- Classification
- Other distributed approaches
  - Web information systems
  - Parallel shared-nothing architectures
  - Distributed data streams
  - Uncertain data
- Open research issues
- Conclusions

Existing Approaches in Unstructured P2P

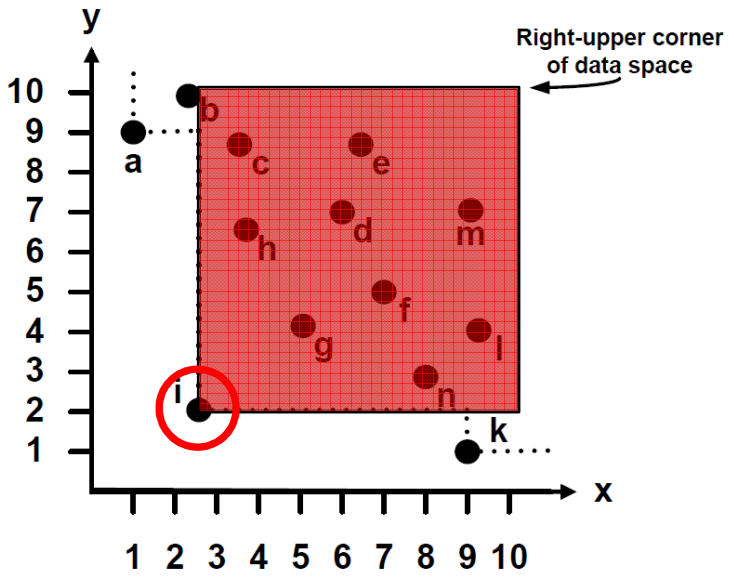
1. **Single Filtering Point (SFP)** [ICDE'06]
2. **DDS** [CIKM'06]
3. **SKYPEER/SKYPEER+** [ICDE'07/ TKDE'10]
4. **BITPEER** [DaMaP'08]
5. **PaDSkyline** [ICDE'08]
6. **SkyPlan** [EDBT'11]
7. **AGiDS** [Globe'09]
8. **FDS** [TKDE'09]

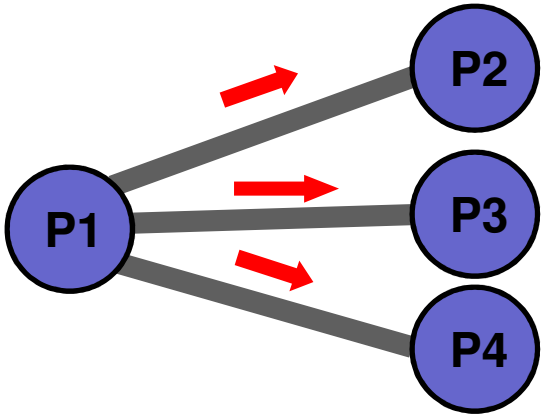
1. Single Filtering Point (SFP)

- Mobile devices communicating via an ad-hoc network (MANETs)
- Skyline queries that involve spatial constraints
- Filtering technique
  - is directly applicable in P2P networks
  - the usage of a local skyline point as a filter to discard local skyline points of other peers

1. Single Filtering Point (SFP)

- Volume of dominating region
  - the area of the data space that is dominated by a skyline point
  - uniform distribution: a higher probability to dominate points
- A peer receives the query
  - processes the query locally
  - propagates the query to its neighboring peers by attaching a filter point to the query



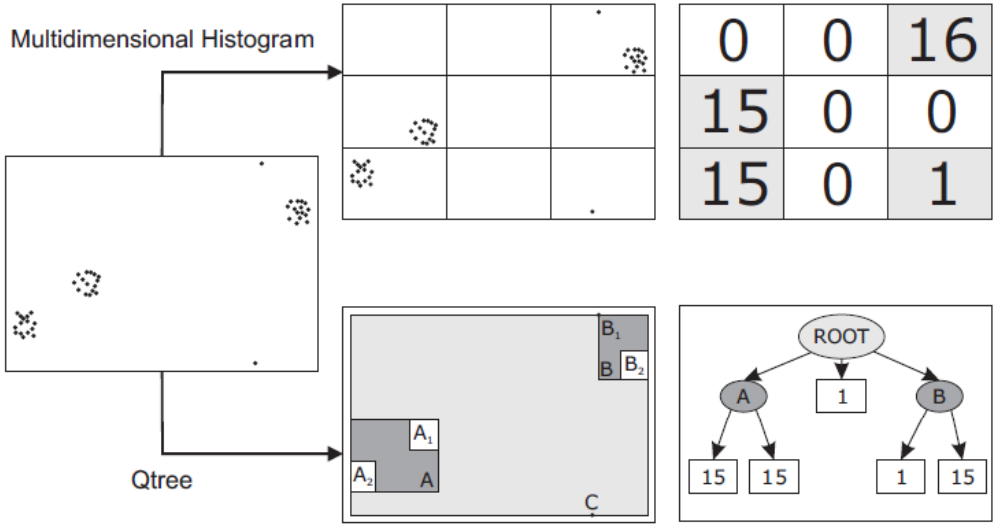


2. DDS

- Uses distributed data summaries (DDS) to identify relevant peers
  - summarize the data accessible via a peer's neighbors
  - one summary for each neighbor summarizing not only its local data but also the data of peers that are located several hops away but reachable via the neighbor
- Two variants of data summaries
  - multidimensional histograms
  - Qtree, a combination of R-trees and multidimensional histograms

Hose *et al.*: Processing Relaxed Skylines in PDMS Using Distributed Data Summaries (CIKM'06)

2. DDS



- Data summaries: neighbors that provide only dominated data are pruned
- The query and the local skyline points are forwarded to the neighboring peers
- Local skyline points are routed back on the same path the query was propagated on

Hose *et al.*: Processing Relaxed Skylines in PDMS Using Distributed Data Summaries (CIKM'06)

2. DDS

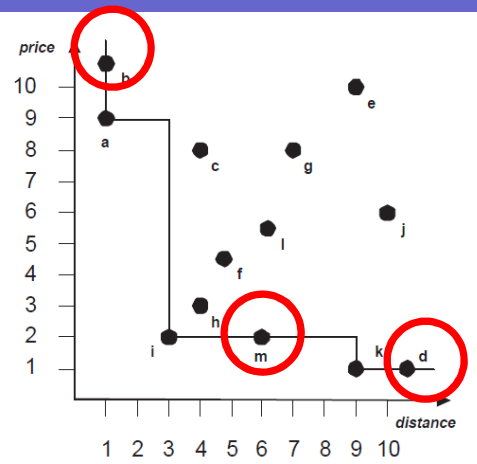
- DDS supports approximate skyline queries (relaxed skylines)
- A relaxed skyline query represents regions of a peer's data by a single local point
- A region describing the data of a neighboring peer is represented by only one local point if any point of the region has a distance to the representative point smaller than a given threshold
- Relaxed skyline
  - does not contain all skyline points
  - additionally representative data points that represent regions that are nearby and possibly contain skyline points

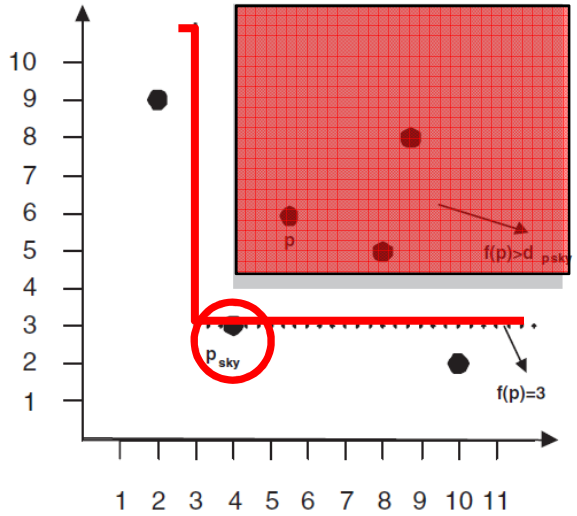
Hose *et al.*: Processing Relaxed Skylines in PDMS Using Distributed Data Summaries (CIKM'06)

3. SKYPEER / SKYPEER+

- Subspace skyline processing over a super-peer architecture
- Each super-peer computes and stores (preprocessing) the extended skyline set of its associated peers
  - the extended skyline set contains all data points that are sufficient to answer a skyline query in any arbitrary subspace
  - data is transformed into one-dimensional values

$$f(p) = \min_{i=1}^{d}(p[i])$$



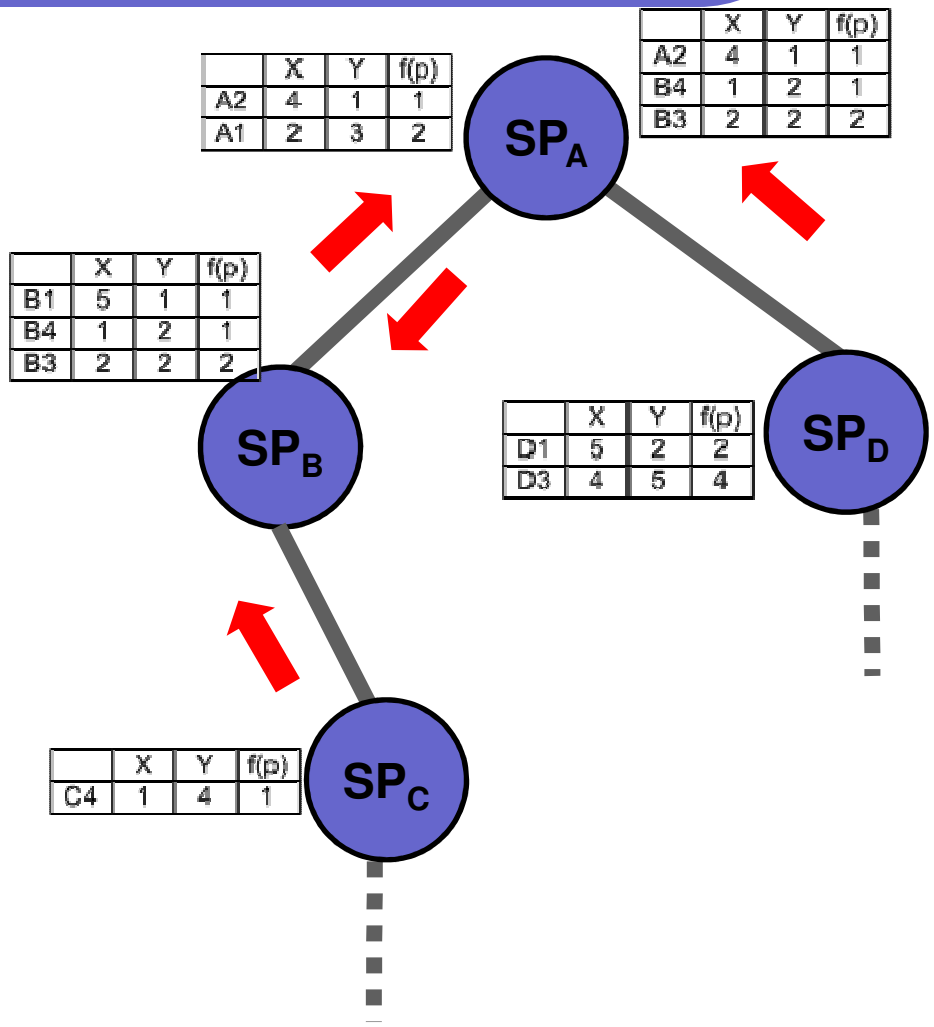


Vlachou *et al.*: Efficient routing of subspace skyline queries over highly distributed data (TKDE'10)

3. SKYPEER / SKYPEER+

- SKYPEER
  - a threshold value
  - all super-peers are queried
- Different strategies for
  - threshold propagation and
  - result merging over the P2P network

  aiming to reduce both computational time and volume of transmitted data
- Threshold propagation:
  - fixed threshold (the query initiator sets the threshold)
  - refined threshold (each super-peer updates the threshold)
- Result merging
  - merging at the query initiator
  - progressive merging



Vlachou *et al.*: Efficient routing of subspace skyline queries over highly distributed data (TKDE'10)

3. SKYPEER / SKYPEER+

- SKYPEER+
  - efficient routing of skyline queries
  - reducing the number of contacted super-peers
- A clustering algorithm (preprocessing) is applied on the locally stored extended skyline set
  - the cluster descriptions are broadcast over the super-peer network
- Each super-peer collects the cluster information of all super-peers and builds routing indexes
  - the one-dimensional mapping is combined with the clustering information (represented by MBRs)
- The routing indexes
  - propagate the query only to network paths with super-peers storing non-dominated points
  - refine the threshold

Vlachou *et al.*: Efficient routing of subspace skyline queries over highly distributed data (TKDE'10)

4. BITPEER

- Subspace skyline queries over a super-peer architecture
- Each super-peer stores the extended skyline and a bitmap representation is used to summarize all extended skyline points
- Given a subspace skyline query
  - the query is flooded in the super-peer network
  - local results are sent back using progressive merging
- Caching of subspace skyline points
  - the querying super-peer gathers the extended subspace skyline instead of the subspace skyline
  - cached results are used for queries that refer to a subspace of the query in the cache

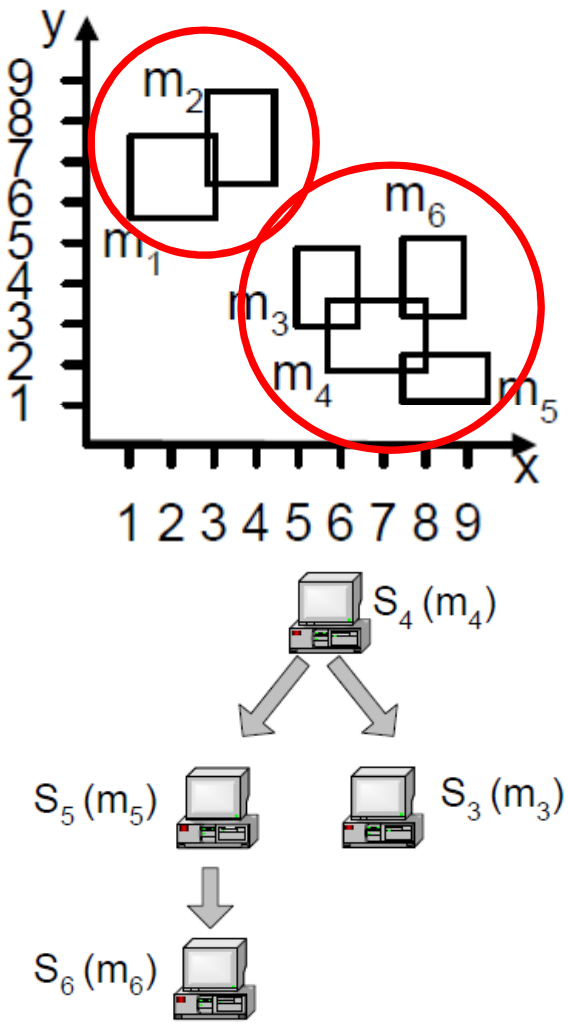
Fotiadou *et al.*: BITPEER: continuous subspace skyline computation with distributed bitmap indexes (DaMap'08)

5. Parallel Distributed Skyline (PaDSkyline)

- The querying peer
  - communicates directly with all peers
  - gathers a set of Minimum Bounding Regions (MBRs) from each peer that summarizes the data stored at each peer
- The main principle is to determine which peers can process the query in parallel and report the results independently

5. Parallel Distributed Skyline (PaDSkyline)

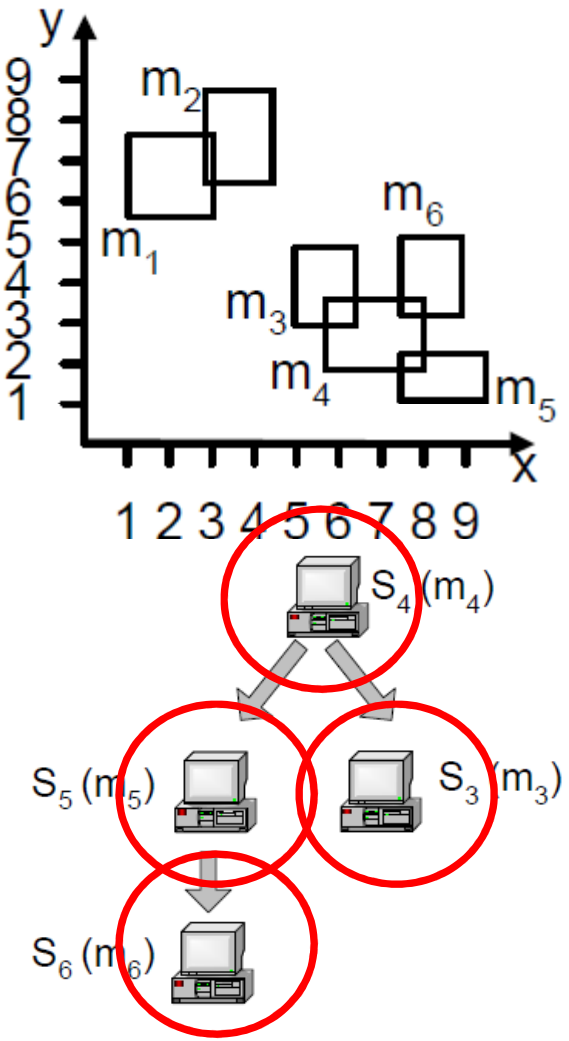
- Dominated MBRs are discarded
- Partially dominated MBRs are executed after the partially dominating MBR
- MBRs are divided into one or more incomparable groups
  - any data point summarized by an MBR of one group cannot be dominated by or dominate data points of another group
- A specific plan is constructed for each incomparable group
- Filter points
  - select the K points with the largest volume of their dominating region
  - pick the K points with the maximal distance between them



Cui *et al.*: Parallel distributed processing of constrained skyline queries by filtering (ICDE'08)

5. Parallel Distributed Skyline (PaDSkyline)

- The plan is sent to the peer (head group) responsible for the head MBR of the plan
- Each peer processes the query locally
  - attaches a set of K filter points
  - removes itself from the query plan
  - forwards the query to the next peer indicated by the plan
  - the results are sent back directly to the head group
- Head group
  - discards all dominated points
  - sends back the results to the querying peer

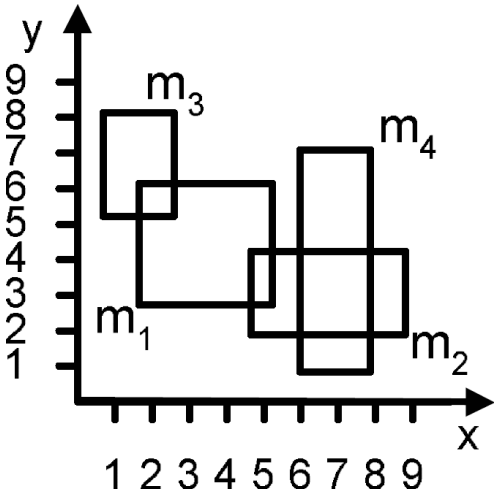


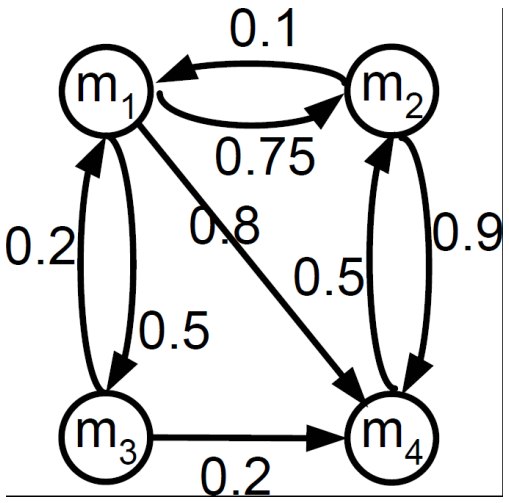
6. SkyPlan

- SkyPlan addresses the problem of generating efficient execution plans
- Querying the peers consecutively
  - the number of queried peers may be reduced
  - the amount of transferred data can be drastically reduced
  - if the filter points fail to prune any point of a peer, then no gain can be obtained from querying the peers consecutively
- Parallelism minimizes the latency

6. SkyPlan

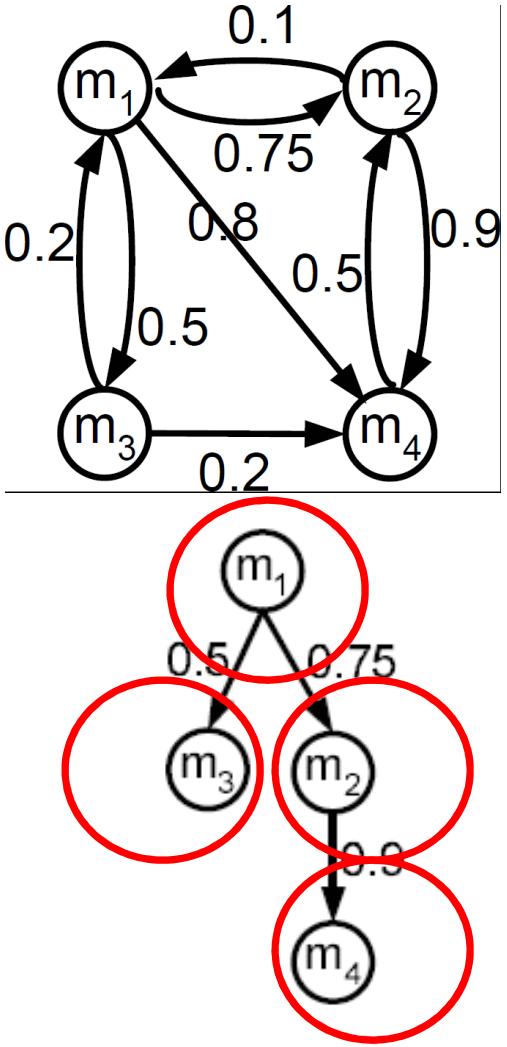
- The query originator creates a weighted directed graph (SD-graph)
  - vertex: a non-dominated MBR
  - an edge: one MBR dominates partially the other MBR
  - the weights: the pruning power
- The SD-graph is transformed into an execution plan (one or more directed trees)
  - maximizes the total pruning power





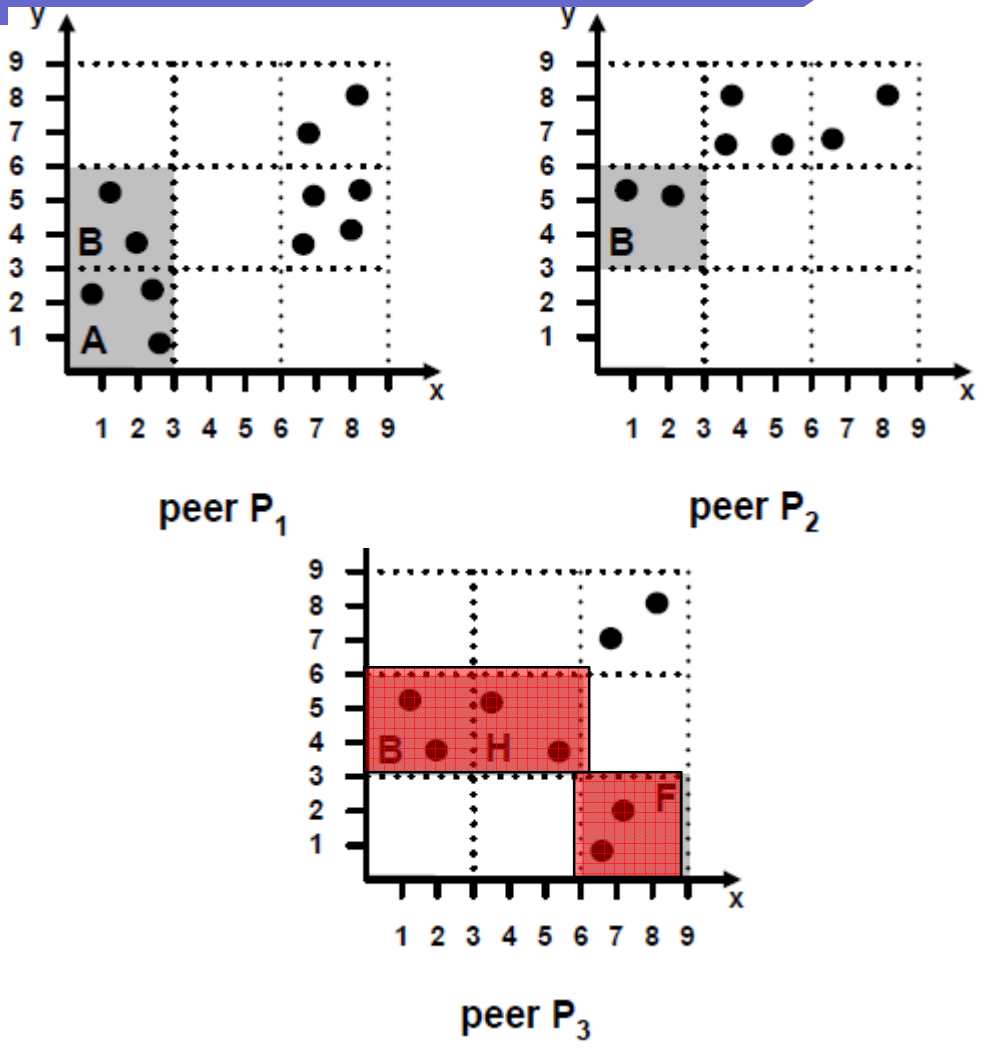
6. SkyPlan

- Each queried peer processes the skyline query locally
  - refines execution plan
  - filter points are selected based on the dominating region by taking into account the MBRs of the execution plan
- Each peer returns the merged skyline points to the previous peer based on the execution plan



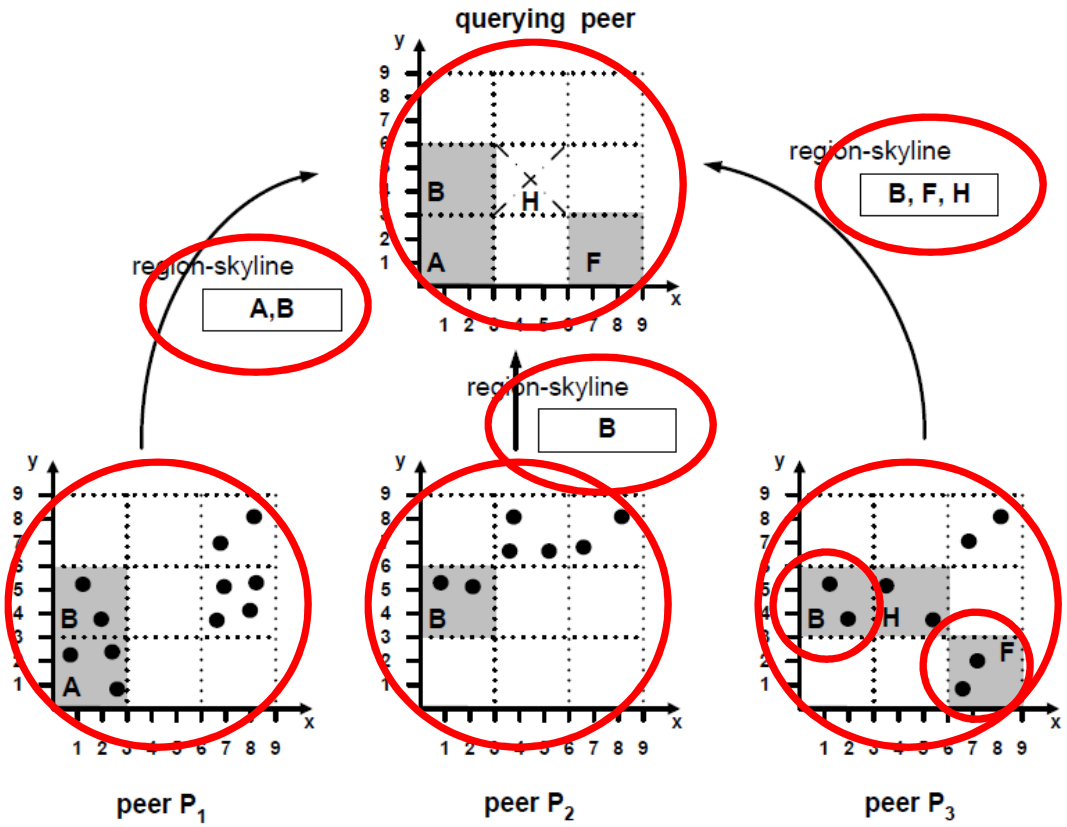
7. AGiDS

- Each peer maintains a grid-based data summary structure
  - all peers share common cell boundaries for the grid structure
  - non overlapping cells
  - efficient merging of local skyline set
- *Region-skyline set of the peer:* the cells that contain at least one data point and that are not dominated by other cells
- Only these cells of the grid contain data that belong to the local skyline set



7. AGiDS

- The region-skyline sets of all peers are collected
- Collected cells are merged into a new region skyline set by discarding dominated cells
- The peers that correspond to cells in the region-skyline set are queried
- Only the local skyline points of the skyline cells are requested
- Result merging by testing only the necessary regions for dominance



Rocha *et al.*: AGiDS: A grid-based strategy for distributed skyline query processing (Globe'09)

8. Feedback-based Distributed Skyline Algorithm (FDS)

- Minimizes the network bandwidth consumption (number of tuples) transmitted over the network
- Each query is processed in multiple round trips
- A scoring function that is used by all peers

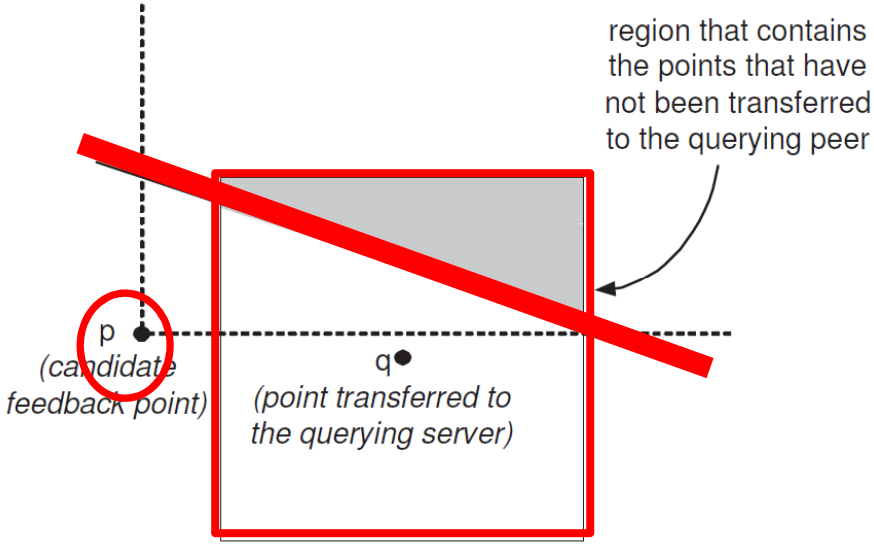
8. Feedback-based Distributed Skyline Algorithm (FDS)

- In each round trip,
  - all peers send the k local skyline points with the lowest score based on the scoring function
  - the querying peer computes the maximum score of all transferred points
  - requests from all peers the local skyline points that have scores smaller than the maximum score
  - merges the local result sets
  - selects some points as feedback
  - peers remove their dominated points
- Filter points are selected for each peer
  - skyline points that dominate at least ℓ local data points

Zhu *et al.*: Distributed skyline retrieval with low bandwidth consumption (TKDE'09)

8. Feedback-based Distributed Skyline Algorithm (FDS)

- The distance of the ℓ-nearest neighbor (L∞) is attached to each local skyline point
  - is combined with the score of the scoring function for selecting feedback
- The depicted rectangle is defined by the distance of the ℓ-nearest neighbor point
- The scoring function defines the region that encloses the unprocessed points
- If the dominating region of a point covers this region, it will dominate at least ℓ points



Zhu *et al.*: Distributed skyline retrieval with low bandwidth consumption (TKDE'09)

Outline

- ***Skyline processing in unstructured P2P***
  - Existing approaches for skyline queries
  - ***Skyline variants***
- Classification
- Other distributed approaches
  - Web information systems
  - Parallel shared-nothing architectures
  - Distributed data streams
  - Uncertain data
- Open research issues
- Conclusions

Skyline variants

- Subspace skyline queries
  - SKYPEER and BITPEER were proposed for
  - PaDSkyline, SkyPlan, and DDS use MBRs for query routing and can easily be adapted
  - AGiDS depends on the grid-based data summary
- Constrained skyline queries
  - DDS and PaDSkyline proposed for
  - SkyPlan and AGiDS can easily be adapted
  - SKYPEER and BITPEER cannot support
- Dynamic skyline queries
  - only approaches with an MBR-based routing mechanism (DDS, SkyPlan and PaDSkyline) may support dynamic skyline queries for some functions

Skyline variants

Approach	Skyline	Subspace	Constrained	Dynamic
DSL [51]	◇		×	
SSP/Skyframe [49, 50]	×		◇	
iSky [9, 12]	×			
SFP [20]	×	◇	◇	◇
DDS [18, 19]	×	◇	×	◇
SKYPEER/SKYPEER+ [45, 46]	◇	×		
BITPEER [16]	◇	×		
PaDSkyline [8, 13]	◇	◇	×	◇
AGiDS [35]	×	◇	◇	
FDS [56]	×	◇	◇	◇
SkyPlan [36]	×	◇	◇	◇

Different skyline query variants supported by distributed approaches (×: proposed for, ◇: also supports)

Outline

- Skyline processing in unstructured P2P
  - Existing approaches for skyline queries
  - Skyline variants
- ***Classification***
- Other distributed approaches
  - Web information systems
  - Parallel shared-nothing architectures
  - Distributed data streams
  - Uncertain data
- Open research issues
- Conclusions

Classification

Approach	Overlay	
DSL [51]	structured	DHT (CAN)
SSP/Skyframe [49, 50]	structured	tree-based (BATON) and DHT (CAN)
iSky [9, 12]	structured	tree-based (BATON)
SFP [20]	unstructured	pure P2P
DDS [18, 19]	unstructured	pure P2P
SKYPEER/SKYPEER+ [45, 46]	unstructured	hybrid (super-peer)
BITPEER [16]	unstructured	hybrid (super-peer)
PaDSkyline [8, 13]	unstructured	fully-connected network topology
AGiDS [35]	unstructured	fully-connected network topology
FDS [56]	unstructured	fully-connected network topology
SkyPlan [36]	unstructured	fully-connected network topology

Classification - Filter Points

- Almost all approaches use the principle of filtering
- Filter points are attached to the query and forwarded to neighboring peers
  - dominated local data points are discarded
  - eliminate neighboring peers that store only dominated data points
- One, multiple, or all local skyline points
  - SSP/Skyframe, iSky, SFP (most dominating point) and SKYPEER/SKYPEER+ (threshold)
  - PaDSkyline, SkyPlan and FDS (multiple)
  - DSL and DDS (all local skylines)
  - BITPEER and AGiDS (none)

Classification - Routing

- A peer tries to eliminate as many neighbors as possible and forwards the query only to the remaining neighbors
- Structured P2P systems
  - a peer exploits the information about data distribution in the underlying overlay (DSL, SSP/Skyframe, iSky)
- Unstructured P2P networks
  - flooding (SKYPEER, BITPEER)
  - routing indexes (DDS, SKYPEER+)
  - exhaustive (PaDSkyline, SkyPlan, AGiDS, FDS)

Classification - Result Merging

- Skyline processing causes queries to travel along paths in the network
- Each peer sends its local result set directly to the query initiator
  - local skyline points have to be sent only once
  - computational load at the initiator is relatively high
  - fully connected network topology (PaDSkyline, AGiDS, FDS)
  - structured overlay (SSP/Skyframe, iSky)
- Local result sets are propagated back using the same path that the query has been forwarded
  - each peer receives the results discards dominated local skyline points
  - P2P network (DDS, SKYPEER/ SKYPEER+, BITPEER)
  - DSL and SkyPlan

Outline

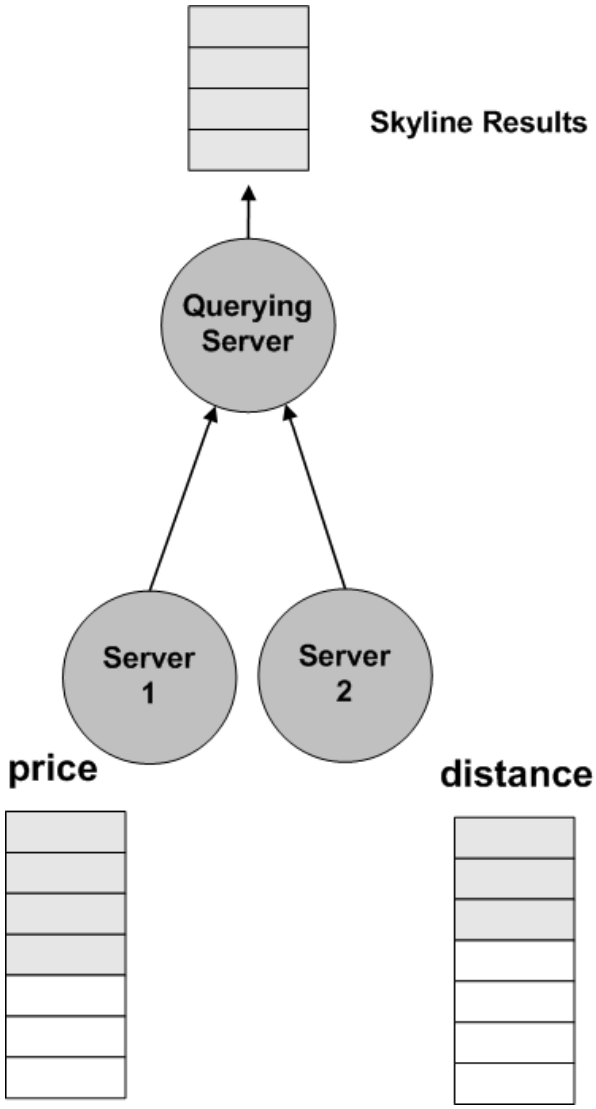
- Skyline processing in unstructured P2P
  - Existing approaches for skyline queries
  - Skyline variants
- Classification
- ***Other distributed approaches***
  - ***Web information systems***
  - Parallel shared-nothing architectures
  - Distributed data streams
  - Uncertain data
- Open research issues
- Conclusions

Web Information Systems

- Each source stores the object identifier and a different attribute of the data objects
- Hotel example:
  - the price is provided by a travel agency
  - the distance to the beach by an online server providing geographical information
- Approaches for web information systems have different objectives than those discussed
- Fundamentally different setup than in highly distributed systems:
  - the number of sources (for each query) is much smaller in comparison to the total number of servers in a highly distributed system
  - query processing aims to minimize the response time, without the restriction of having a fixed number of round-trips
  - all sources need to be contacted

Web Information Systems

- Two basic methods of data access are provided
  - sorted access: retrieving the next object with the best value with respect to a single attribute
  - random access: retrieving the attribute value for a certain given object
- Three algorithms have been proposed
  - basic distributed skyline algorithm (BDS)
    Balke *et al.* @ EDBT'04
  - improved distributed skyline algorithm (IDS)
    Balke *et al.* @ EDBT'04
  - progressive distributed skyline algorithm (PDS)
    Lo *et al.* @ DKE'06



Web Information Systems

- All algorithms consist of two phases
- First phase
  - a subset of objects is retrieved that includes at least all skyline objects
  - sorted access from the different sources until all attribute values of at least one data object have been retrieved from all sources
- Second phase
  - discards all the non-skyline objects in the subset by pruning dominated points.
  - missing attribute values that are required for the domination tests are retrieved through random access
  - the number of domination tests is reduced: an object o can be dominated only by objects that have been retrieved from the same sources as o

Web Information Systems

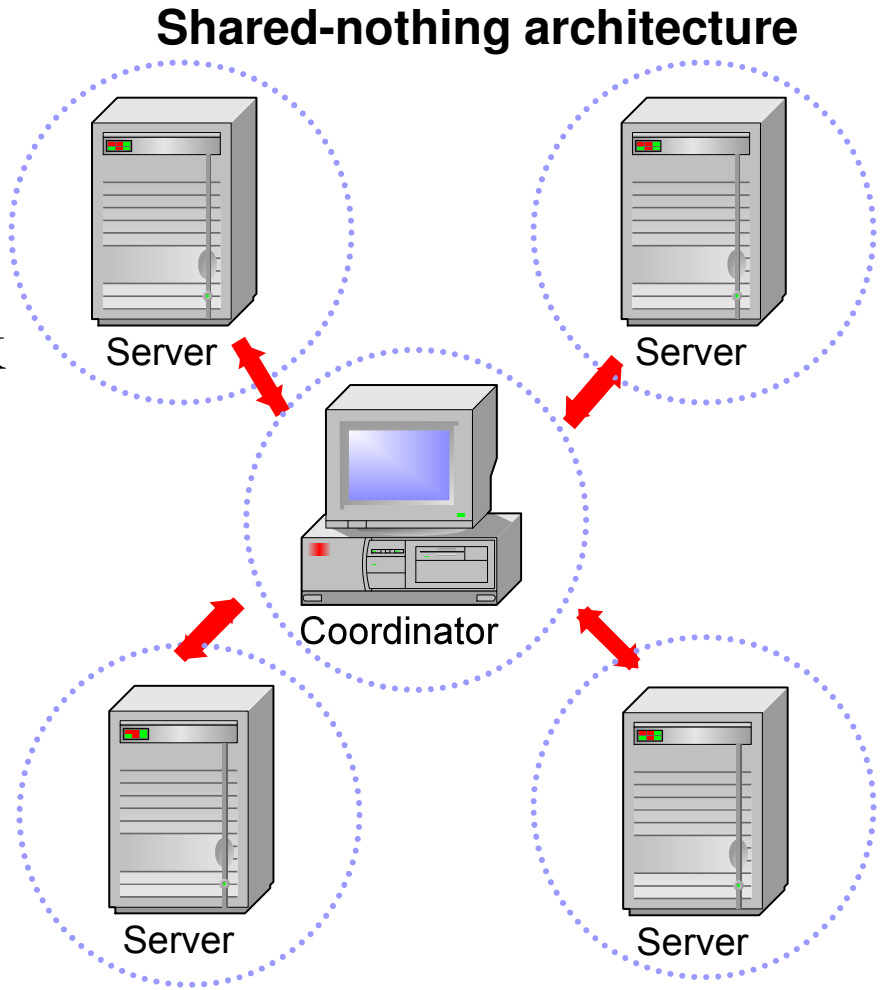
- An important differentiating feature of the three algorithms is the order in which the sources are accessed during the first phase
  - influences the efficiency of the algorithm
  - influences the number of sorted accesses
  - leads to a terminating object with fewer accesses
- BDS: each data source is accessed in a round-robin fashion
- IDS: uses a heuristic to detect the most promising source
  - estimates the remaining number of sorted accesses required to retrieve all missing attributes
  - the difference between the missing attribute values and the last values retrieved through sorted access from each source
  - requires that the missing attribute values of the most probable terminating object are retrieved through random access
- PDS: uses a linear-regression method to estimate the ranks of the object and determine a better order to access the sources

Outline

- Skyline processing in unstructured P2P
  - Existing approaches for skyline queries
  - Skyline variants
- Classification
- ***Other distributed approaches***
  - Web information systems
  - ***Parallel shared-nothing architectures***
  - Distributed data streams
  - Uncertain data
- Open research issues
- Conclusions

Parallel Shared-Nothing Architecture

- Queries arrive at a central node, i.e. coordinator server
- Skyline processing is CPU-intensive: the coordinator distributes the processing task to all available servers
- The input data is partitioned and each partition is assigned to one server
- The query is executed simultaneously on all servers
- The local results are
  - send back to the coordinator
  - merged to the skyline set

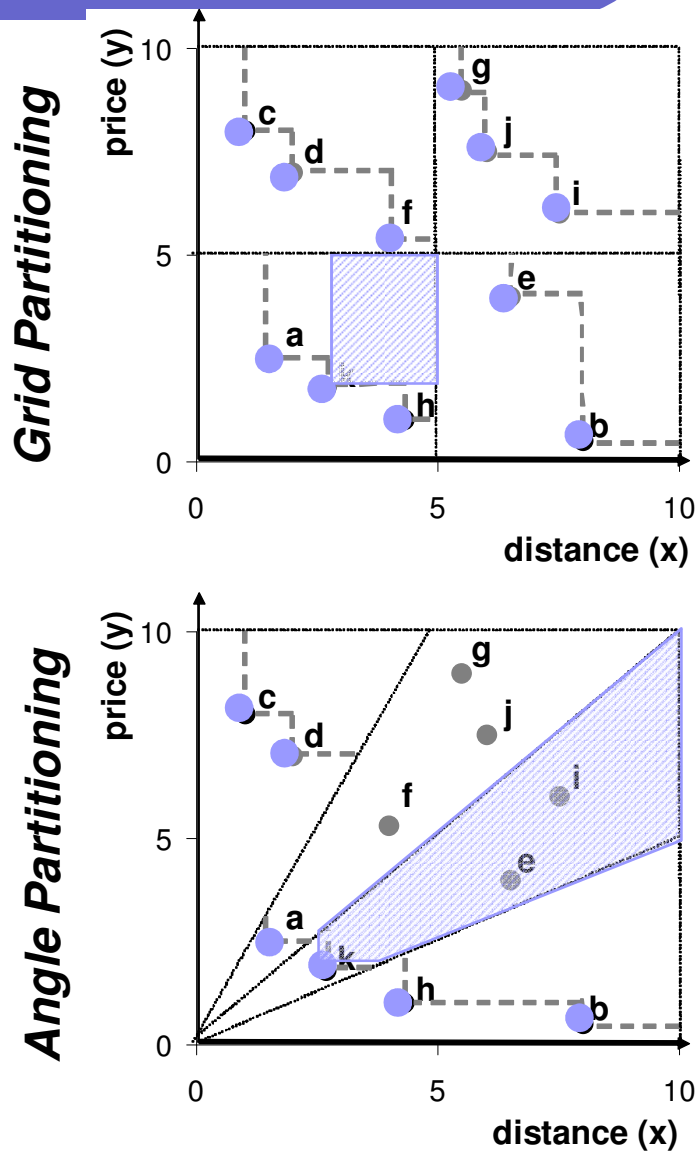


Parallel Shared-Nothing Architecture

- Performance of the parallel skyline computation
  - local skyline computation
  - amount of transferred local skyline points
  - performance of merging phase
- The goal is to minimize response time
  - sharing the workload evenly among all participating servers
  - approximately the same number of data points
  - the skyline algorithm should have similar performance on the data points in every partition
  - the local skyline points returned to the coordinator for the merging phase should be minimized
- Important factor: space partitioning method used for distributing the dataset among the servers

Parallel Shared-Nothing Architecture

- Partitioning schemes: random, grid and angle
- Grid-based partitioning:
  - 11 local skyline points
- Angle-based partitioning:
  - 6 local skyline points
  - 5 of them in the skyline set
- Higher pruning power of local skyline points for the angle partitioning



Outline

- Skyline processing in unstructured P2P
  - Existing approaches for skyline queries
  - Skyline variants
- Classification
- ***Other distributed approaches***
  - Web information systems
  - Parallel shared-nothing architectures
  - ***Distributed data streams***
  - Uncertain data
- Open research issues
- Conclusions

Distributed Data Streams

- Each data object is associated with a timestamp indicating its time of arrival
- The width of the sliding window defines the lifespan of any object
- Given a timestamp, the skyline set contains the data objects that are valid at this timestamp and not dominated by any other valid data object at that timestamp
- Centralized algorithms for skyline queries over streams focus on efficiently detecting data objects that become skyline points after a skyline point expires

Distributed Data Streams

- BOCS (Sun *et al.* @ KIS'10) relies on distributed data stream model
  - servers are producing the data objects of the stream
  - a central server that communicates with the remote servers
  - responsible for evaluating the queries
- The challenge is to efficiently monitor the skyline over time, rather than computing the skyline at a given timestamp
  - each server monitors the local skyline set (centralized algorithm)
  - objects that are added to the skyline set are sent to the central server
  - the central server applies a centralized skyline algorithm over the received data to compute the skyline set at each timestamp

Outline

- Skyline processing in unstructured P2P
  - Existing approaches for skyline queries
  - Skyline variants
- Classification
- ***Other distributed approaches***
  - Web information systems
  - Parallel shared-nothing architectures
  - Distributed data streams
  - ***Uncertain data***
- Open research issues
- Conclusions

Uncertain Data

- Skyline probability of a data object
  - the probability that this data object exists, while all data objects that dominate it do not exist
- A probabilistic skyline query is associated with a given threshold
  - all data objects with a skyline probability higher than the threshold

Uncertain Data

- Processing a distributed skyline query over uncertain data
  - each server computes its local probabilistic skyline set
  - result merging: the probability is refined based on all collected data points
  - points with skyline probabilities smaller than the threshold are discarded
- The property of additivity does not hold for the probabilistic skyline query
  - candidate skyline points are sent to all servers in order to compute their exact probabilities on all data points
- The correctness of this approach
  - the local probability is an upper bound of its actual probability
  - the probabilities can be computed accumulatively

Outline

- Skyline processing in unstructured P2P
  - Existing approaches for skyline queries
  - Skyline variants
- Classification
- Other distributed approaches
  - Web information systems
  - Parallel shared-nothing architectures
  - Distributed data streams
  - Uncertain data
- ***Open research issues***
- Conclusions

Open Research Directions

- None of the existing approaches has studied dynamic skyline queries
  - more challenging as the skyline computation is based on a set of user specified functions
  - hotel example: each hotel is associated with its geographical coordinates, while the user tries to minimize the distance to a location of interest
- Continuous skyline maintenance
  - try to keep the skyline set up-to-date in the presence of data insertion or deletions

Open Research Directions

- Only limited work exists for distributed probabilistic skyline queries
- In distributed environments, the uncertainty of the data occurs naturally
  - the data itself can be uncertain (in sensor networks, the uncertainty can be the result of noisy readings from sensors)
  - peers themselves may not always be trusted by other peers and some of them might act as cheaters deliberately

Open Research Directions

- The cardinality of the skyline set can be high
  - high processing cost
  - high bandwidth consumption
  - the total number of local skyline points is higher than the number of skyline points
  - in mobile networks or cloud computing, the users may be charged based on the amount of transferred data
- Cardinality estimation of the skyline set is very important in distributed environments
- Approximation of the skyline set by selecting only a subset of the local skyline sets
  - representative skyline points
  - ranking the skyline points

Outline

- Skyline processing in unstructured P2P
  - Existing approaches for skyline queries
  - Skyline variants
- Classification
- Other distributed approaches
  - Web information systems
  - Parallel shared-nothing architectures
  - Distributed data streams
  - Uncertain data
- Open research issues
- ***Conclusions***

Conclusions

- Data is increasingly stored in distributed way, therefore distributed query processing is an important problem
- Skyline operator: identifies a set of interesting objects in a large database
- We outlined the main principles of distributed skyline processing and surveyed existing approaches
- There are still interesting and challenging issues about distributed skyline processing that have not been studied so far in the related literature

Thank you!

Katja Hose, Akrivi Vlachou:
"A Survey of Skyline Processing in Highly Distributed Environments"
to appear in VLDB Journal

More information: http://www.idi.ntnu.no/~vlachou/

NTNU
Norwegian University of
Science and Technology

Literature

1. Balke, W., Güntzer, U., Zheng, J.X.: Efficient distributed skylining for web information systems. In: Proc. of International Conference on Extending Database Technology (EDBT), pp. 256–273 (2004)
2. Börzsönyi, S., Kossmann, D., Stocker, K.: The skyline operator. In: Proc. of International Conference on Data Engineering (ICDE), pp. 421–432 (2001)
3. Chan, C.Y., Jagadish, H.V., Tan, K.L., Tung, A.K.H., Zhang, Z.: On high dimensional skylines. In: Proc. of International Conference on Extending Database Technology (EDBT), pp. 478–495 (2006)
4. Chen, L., Cui, B., Lu, H.: Constrained skyline query processing against distributed data sites. IEEE Transactions on Knowledge and Data Engineering (TKDE) 23(2), 204–217 (2011)
5. Chen, L., Cui, B., Lu, H., Xu, L., Xu, Q.: iSky: Efficient and progressive skyline computing in a structured P2P network. In: Proc. of the International Conference on Distributed Computing Systems (ICDCS), pp. 160–167 (2008)
6. Crespo, A., Garcia-Molina, H.: Routing indices for peer-to peer systems. In: Proc. of the International Conference on Distributed Computing Systems (ICDCS), pp. 23–30 (2002)
7. Cui, B., Chen, L., Xu, L., Lu, H., Song, G., Xu, Q.: Efficient skyline computation in structured peer-to-peer systems. IEEE Transactions on Knowledge and Data Engineering (TKDE) 21(7), 1059–1072 (2009)
8. Cui, B., Lu, H., Xu, Q., Chen, L., Dai, Y., Zhou, Y.: Parallel distributed processing of constrained skyline queries by filtering. In: Proc. of International Conference on Data Engineering (ICDE), pp. 546–555 (2008)
9. Ding, X., Jin, H.: Efficient and progressive algorithms for distributed skyline queries over uncertain data. IEEE Transactions on Knowledge and Data Engineering (TKDE) to appear (2011)
10. Fotiadou, K., Pitoura, E.: BITPEER: continuous subspace skyline computation with distributed bitmap indexes. In: Proc. of International Workshop on Data Management in Peer-to-Peer Systems (DaMaP), pp. 35–42 (2008)
11. Hose, K., Lemke, C., Sattler, K.: Processing Relaxed Skylines in PDMS Using Distributed Data Summaries. In: Proc. of International Conference on Information and Knowledge Management (CIKM), pp. 425–434 (2006)
12. Hose, K., Lemke, C., Sattler, K., Zinn, D.: A relaxed but not necessarily constrained way from the top to the sky. In: Proc. of International Conference on Cooperative Information Systems (CoopIS), pp. 339–407 (2007)
13. Huang, Z., Jensen, C.S., Lu, H., Ooi, B.C.: Skyline queries against mobile lightweight devices in manets. In: Proc. of International Conference on Data Engineering (ICDE), p. 66 (2006)

Literature

14. Jagadish, H., Ooi, B., Vu, Q.: BATON: a balanced tree structure for peer-to-peer networks. In: Proc. of International Conference on Very Large Data Bases (VLDB), pp. 661–672 (2005)
15. Lin, X., Yuan, Y., Wang, W., Lu, H.: Stabbing the sky: Efficient skyline computation over sliding windows. In: Proc. of International Conference on Data Engineering (ICDE), pp. 502–513 (2005)
16. Lo, E., Yip, K.Y., Lin, K.I., Cheung, D.W.: Progressive skylining over web-accessible databases. Data Knowledge Engineering (DKE) 57(2), 122–147 (2006)
17. Ratnasamy, S., Francis, P., Handley, M., Karp, R., Schenker, S.: A scalable content-addressable network. In: Proc. of Conference on Applications, technologies, architectures, and protocols for computer communications (SIGCOMM), pp. 161–172 (2001)
18. Risson, J., Moors, T.: Survey of research towards robust peer-to-peer networks: Search methods. Computer Networks 50(17), 3485–3521 (2006)
19. Rocha-Junior, J.B., Vlachou, A., Doulkeridis, C., Nørvåg, K.: AGiDS: A grid-based strategy for distributed skyline query processing. In: Proc. of International Conference on Data Management in Grid and Peer-to-Peer Systems (Globe), pp. 12–23 (2009)
20. Rocha-Junior, J.B., Vlachou, A., Doulkeridis, C., Nørvåg, K.: Efficient execution plans for distributed skyline query processing. In: Proc. of International Conference on Extending Database Technology (EDBT), pp. 271–282 (2011)
21. Stoica, I., Morris, R., Karger, D., Kaashoek, M.F., Balakrishnan, H.: Chord: A scalable peer-to-peer lookup service for internet applications. In: Proc. of Conference on Applications, technologies, architectures, and protocols for computer communications (SIGCOMM), pp. 149–160 (2001)
22. Sun, S., Huang, Z., Zhong, H., Dai, D., Liu, H., Li, J.: Efficient monitoring of skyline queries over distributed data streams. Knowledge and Information Systems 25, 575–606 (2010)
23. Vlachou, A., Doulkeridis, C., Kotidis, Y.: Angle-based space partitioning for efficient parallel skyline computation. In: Proc. of International Conference on Management of Data (SIGMOD), pp. 227–238 (2008)
24. Vlachou, A., Doulkeridis, C., Kotidis, Y., Vazirgiannis, M.: SKYPEER: Efficient subspace skyline computation over distributed data. In: Proc. of International Conference on Data Engineering (ICDE), pp. 416–425 (2007)
25. Vlachou, A., Doulkeridis, C., Kotidis, Y., Vazirgiannis, M.: Efficient routing of subspace skyline queries over highly distributed data. IEEE Transactions on Knowledge and Data Engineering (TKDE) 22(12), 1694–1708 (2010)

Literature

26. Vlachou, A., Nørvåg, K.: Bandwidth-constrained distributed skyline computation. In: Proc. of the International Workshop on Data Engineering for Wireless and Mobile Access (MobiDE), pp. 17–24 (2009)
27. Wang, J., Wu, S., Gao, H., Li, J., Ooi, B.C.: Indexing multidimensional data in a cloud system. In: Proc. of International Conference on Management of Data (SIGMOD), pp. 591–602 (2010)
28. Wang, S., Ooi, B., Tung, A., Xu, L.: Efficient skyline query processing on peer-to-peer networks. In: Proc. of International Conference on Data Engineering (ICDE), pp. 1126– 1135 (2007)
29. Wang, S., Vu, Q.H., Ooi, B.C., Tung, A.K., Xu, L.: Skyframe: a framework for skyline query processing in peer-to-peer systems. The VLDB Journal 18(1), 345–362 (2009)
30. Wu, P., Zhang, C., Feng, Y., Zhao, B., Agrawal, D., Abbadi, A.: Parallelizing skyline queries for scalable distribution. In: Proc. of International Conference on Extending Database Technology (EDBT), pp. 112–130 (2006)
31. Yang, B., Garcia-Molina, H.: Designing a super-peer network. In: Proc. of International Conference on Data Engineering (ICDE), pp. 49–60 (2003)
32. Zhang, Z., Yang, Y., Cai, R., Papadias, D., Tung, A.: Kernelbased skyline cardinality estimation. In: Proc. of International Conference on Management of Data (SIGMOD), pp. 509–522 (2009)
33. Zhu, L., Tao, Y., Zhou, S.: Distributed skyline retrieval with low bandwidth consumption. IEEE Transactions on Knowledge and Data Engineering (TKDE) 21(3), 384–400 (2009)