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

Katja Hose
Max-Planck Institute for Informatics
Saarbrücken, Germany
hose@mpi-inf.mpg.de

Akrivi Vlachou
Dept. of Computer and Information Science
Norwegian University of Science and Technology
Trondheim, Norway
vlachou@idi.ntnu.no

ABSTRACT

During the last decade, data management and storage have become increasingly distributed. In consideration of the huge amount of data available in such systems, advanced query operators, such as skyline queries, are necessary to help users process the data. For example, a user who is interested in buying a car wants to find a good trade-off between minimum age and minimum price. It is not obvious how much cheaper a car should be, if it is one year older than another car. Thus, the skyline query will retrieve a set of data items that are the best trade-offs for the user's preferences. The skyline operator has been proposed about a decade ago, but research on skyline queries, especially in distributed scenarios, is still an ongoing process.

Query processing in distributed environments poses inherent challenges and requires non-traditional techniques due to the distribution of content and the lack of global knowledge. In this tutorial, we will outline the objectives and the main principles that any distributed skyline approach has to fulfill, leading to useful guidelines for the design of efficient distributed skyline algorithms. More importantly, distributed processing of other query types share the same objectives and principles, therefore several of the guidelines are applicable also for other query types. Furthermore, this tutorial will provide a broad survey of the state-of-the-art in distributed skyline processing, present a categorization of the existing approaches based on their characteristics, and point out open research challenges in distributed skyline processing.

Categories and Subject Descriptors

H.2.4 [**Database Management**]: Systems—*Query Processing, Distributed Databases*

General Terms

Algorithms, Experimentation, Performance

Keywords

Skyline Query, Distributed Systems, P2P, Query Processing

EDBT 2012 March 26–30, 2012, Berlin, Germany

1. INTRODUCTION

Developments in the past couple of years have revealed a trend towards distributed data management and storage systems. In the presence of the huge amounts of data that today's systems are providing access to, it is a tedious task for a user to find the most interesting available data without using advanced query types, such as skyline queries. Whereas the problem is known as skylines in database research, in other areas it was already known before as the maximum vector problem or the Pareto optimum [11, 15]. The popularity of the skyline operator is mainly due to its applicability for decision making applications; skyline queries help users make intelligent decisions over complex data, where different and often conflicting criteria are considered.

Consider, for instance, a database containing information about hotels. Assume a user is looking for hotels at a specific location that are as cheap as possible and as close as possible to the beach. In this case, it is not obvious whether the user would prefer a hotel that is very close to the beach but more expensive than others or rather a hotel that is very cheap but farther away from the beach. The skyline set contains all hotels that are not worse than any other hotel based on all criteria, without requiring a scoring function that defines the relative importance of the different criteria. Thus, the skyline set contains all tuples that represent the best trade-offs between the different criteria. Figure 1 shows an example, where each point represents a hotel with price per night and distance to the beach as coordinates; hotels a, i, m, and k are in the skyline set.

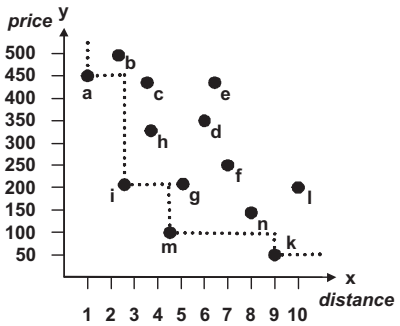

Figure 1: Skyline example

The same considerations also hold for a variety of applications (e.g., electronic marketing places or real-estate databases for houses), where the user is interested in mobiles, cars, houses, or other products. The user might, for instance, be looking for a new mobile sup-

porting all fancy features such as WiFi, high resolution camera and display, but with long talk/standby times and still minimum weight and size. Likewise, a user who is interested in buying a car wants to find a good trade-off between minimum mileage, minimum age, and minimum price.

Skyline queries have originally been proposed for centralized environments [1], i.e., single-database environments. As nowadays data is increasingly stored and processed in a distributed way, skyline processing over distributed data has attracted much attention recently. Consider a global-scale web-based hotel reservation system, consisting of a large set of independent servers geographically dispersed around the world (Figure 2). Servers accept subscriptions from travel agencies in order to advertise their hotels. Each server may provide offers for hotels all over the world, e.g., servers in Paris, Sydney, and Los Angeles might provide different offers for hotels in Miami. Such a system could potentially provide booking services without requiring each travel agency to register with each server. This need becomes even more important due to the fact that the number of providers (and therefore data) increases at tremendous rates. The challenge is to enable users to pose interesting queries, such as skyline queries, over a network of servers and retrieve only those tuples that match the user-defined query.

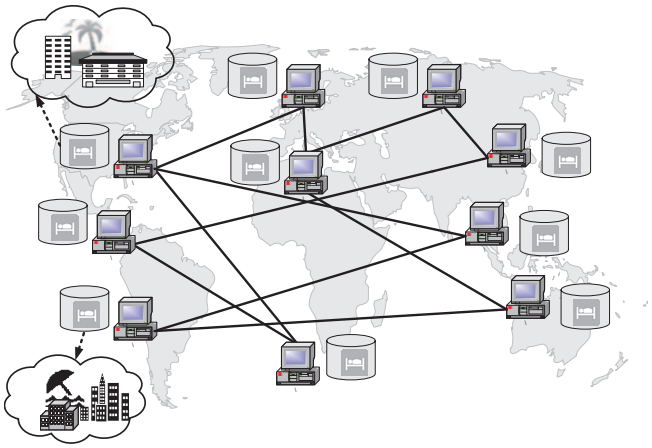

Figure 2: Distributed hotel reservation system

Skyline query processing in distributed environments poses inherent challenges and requires non-traditional techniques due to the distribution of content and the lack of global knowledge. There are various different distributed systems with different requirements and unique characteristics that have to be exploited for efficient skyline processing. Peer-to-peer (P2P) systems can be considered as an example of a distributed system architecture for which several distributed skyline approaches have been proposed. Other architectures, such as Web information systems, parallel shared-nothing architectures, distributed data streams, or wireless sensor networks have different requirements, therefore specialized algorithms have been proposed.

The variety of existing distributed systems leads to a variety of existing distributed skyline approaches. Moreover, the fact that several skyline variants, beyond the traditional skyline operator, have also been proposed in the past decade [5,13,14,22,23] leads to various distributed approaches that support different skyline variants. The most important variants are: subspace skylines (only some attributes of a tuple are considered for evaluation), constrained skylines (only tuples that have values in a given range are considered, e.g., tuples with the distance to the beach in $[0, 10]$ and the price per night in $[50, 200]$), and dynamic skyline queries (the skyline is not executed in the original data space but the data points are transformed into another data space before evaluating the skyline). The characteristics of the skyline variants require sophisticated and specialized algorithms for efficient processing. Depending on the underlying network and communication architecture, these variants allow for different optimizations.

In recent years, there has been much research on distributed skyline query processing so that today it is hard to get an overview of this area. The goal of this tutorial is to provide an overview of the state-of-the-art without requiring expert knowledge on the topic. We will outline the objectives and the main principles that any distributed skyline approach has to fulfill, leading to useful guidelines for developing algorithms for distributed skyline processing. We will discuss in detail existing approaches, clarify the assumptions of each approach, and provide a comparative performance analysis. Our analysis leads to a taxonomy of existing approaches and a categorization based on the skyline variants each approach supports. Finally, we will present interesting research topics on distributed skyline computation that have not yet been explored.

2. OBJECTIVES AND SCOPE

The aim of this tutorial is to provide a broad survey of the state-of-the-art in distributed skyline processing as well as to lead to a deeper understanding of the existing approaches and point out open research challenges in distributed query processing. The tutorial is based on a recent survey [9].

The skyline operator was introduced [1] for a single database environment, i.e., in a centralized setup. Since its introduction in 2001, over a hundred papers have been published. These papers have not only studied efficient skyline computation in centralized or distributed systems but also proposed variations of the traditional skyline operator and studied different premises. Nowadays, it is very common that data is stored and processed in a distributed way, therefore skyline processing over distributed data has attracted much attention recently.

In this tutorial, we give an overview of the existing approaches for skyline query processing in highly distributed environments, where each server stores a fraction of the available data. For example, the peer-to-peer (P2P) paradigm emerges as a powerful model for organizing and searching large data repositories distributed over independent sources. Although the majority of relevant papers have been proposed for P2P architectures, many principles are also applicable in other distributed systems, where the data is distributed over autonomous servers such as grid systems, large-scale data centers, or cloud computing infrastructures.

The key property of the skyline operator that is used in distributed skyline processing is the additivity of the skyline operator. Given a set of servers that store locally relevant data, the skyline sets are the same if the skyline operator is evaluated on (i) the union of the local datasets or (ii) first on each dataset in separate and then once more on the union of the result sets. Due to the additivity of the skyline operator, the skyline query can be processed in a distributed fashion, where each queried server processes a skyline query based on the data that are stored locally and reports back its local skyline set (or a fraction of it). An important factor that influences the performance of a distributed skyline approach is query routing, which is the process of deciding which servers may contribute to the skyline set and hence which servers should be queried in the subsequent round.

During query processing, a server aims to forward the query only to the (neighboring) servers that may contribute to the skyline set. Most approaches for distributed skyline processing cause queries

to travel along paths in the network. In general, these paths are not determined in advance but influenced by the local skyline points retrieved at each server, which are used to decide on the subset of neighboring servers that can contribute to the skyline set. For example, in the case of structured P2P systems, a server exploits the information about data distribution in the underlying overlay to route queries efficiently to relevant neighboring servers [3, 4, 12, 20–22]. On the other hand, in unstructured P2P networks due to the absence of any information about the data distribution, a straightforward alternative is to forward the query to all available servers using flooding [6, 10, 18]. Then, each server that receives the query forwards it to all of its neighbors. A more efficient alternative used in unstructured P2P networks is to build routing indexes that store sufficient information to decide on the relevance of neighbors [7, 8, 19]. Finally, in approaches [2, 5, 16, 17, 24] assuming direct communication between servers, the querying server contacts all servers to gather some summary information about their data. Then, based on this information, the querying server decides which servers are queried in a subsequent round in order to compute the skyline set.

The goal of this tutorial is to provide a deep understanding of the principles of distributed skyline processing and provide useful guidelines for the design of efficient distributed algorithms. To this end, we will present an overview of the state-of-the-art in distributed skyline processing and analyze in details the main properties of each approach. The existing approaches are classified based on their characteristics and based on the skyline variants they support. The similarities and differences of the different approaches will be discussed and a comparative performance analysis that relies on the aforementioned differences and similarities will be provided.

3. TUTORIAL OUTLINE

This tutorial is divided into six parts that cover the background knowledge of the area, a survey of the existing techniques, as well as the open research problems. In the following, we provide a short description of the content of each part.

3.1 Introduction

Our tutorial will start with a short introduction to the topic. We will highlight application scenarios for distributed skyline queries proposed in the literature.

3.2 Basic Concepts and Background

In this part of the tutorial, we provide all the required background knowledge for attendees that are unfamiliar with skyline queries or distributed systems.

Skyline Queries: For attendees having little prior knowledge about skyline queries, we will give a short introduction of skyline queries consisting of illustrative examples in addition to formal definitions of the skyline set and its variants (subspace, constrained, and dynamic skylines). To give the audience a basic understanding of skyline processing, we will briefly discuss a basic approach to evaluate skyline queries in centralized environments. Furthermore, we will summarize the evolution of skyline queries in database research over time and shortly describe related work.

Distributed Systems: To provide non-experts in the audience with some background on distributed systems and to point out the most important characteristics to experts, we will also briefly present different architectures of distributed systems focusing on highly distributed systems, namely P2P systems.

3.3 Objectives and Principles of Distributed Skyline Processing

In this part of the tutorial, we will point out the goals that any distributed approach for query processing must fulfill. This will offer a basic understanding of the main challenges related to distributed query processing. Then, we will elaborate on the characteristics of the skyline operator and point out specific optimization techniques that are only applicable to skyline computation in specific environments. Then, we will present the main principles of distributed skyline processing and point out the major phases of any distributed skyline algorithm. We will also describe the objectives of distributed skyline processing and show how these are fulfilled through the aforementioned phases. Thus, we will provide useful guidelines for the design of efficient distributed skyline algorithms. More importantly, distributed processing of other query types share the same objectives and principles, therefore several of the guidelines are applicable also for other query types.

3.4 Approaches for Skyline Processing in Distributed Environments

The main part of the tutorial will then outline in detail existing approaches for processing skylines in highly distributed environments. We will discuss differences and similarities between them and in addition to explaining the algorithms as proposed in the original papers, we will also point out if the presented approach is extensible to support skyline variants other than the approach was originally proposed for. As it is hard to evaluate approaches relying on different system architectures against each other in a fair way, our discussion will also provide a theoretical performance analysis so that the approaches become comparable.

Finally, we will provide a categorization of distributed skyline approaches in a taxonomy that summarizes and highlights their differences and similarities. The categorization is based on the main techniques that distributed approaches use to optimize skyline computation. The main categories are determined by (i) the routing technique that is used, (ii) how are results propagated back to the query initiator, and (iii) whether filter points are used to distinguish relevant and irrelevant peers and paths. We will also briefly discuss the optimization goals, e.g., response time or scalability, that the approaches were designed for.

3.5 Other Distributed Environments

Whereas the main part of the tutorial focuses on highly distributed systems, the tutorial also sketches some approaches that were proposed for other distributed environments. In particular, skyline approaches have been proposed for Web Information Systems, parallel shared-nothing architectures, distributed data streams, and wireless sensor networks. This tutorial will shortly describe these approaches and highlight their fundamental differences to approaches proposed for highly distributed systems.

3.6 Concluding Remarks and Open Problems

We will conclude the tutorial by summarizing the existing work in distributed skyline processing. This summarization allows us to identify interesting and challenging issues about distributed skyline computation that have not been studied so far in the related literature. Thus, this tutorial will end with a discussion of important open problems and research directions that aim to encourage even more researchers to study distributed query processing.

4. INTENDED AUDIENCE

Our tutorial is intended to benefit researchers who are interested in distributed query processing in general. Therefore, we will point

out general objectives and principles applicable not only for skyline queries but also for many other query types. Thus, we expect that the majority of audience will be familiar with basic knowledge about the general concepts of distributed systems but with little prior knowledge on skyline queries. Nevertheless, our tutorial provides the necessary background so that non-experts in distributed processing can follow the tutorial. For attendees having little prior knowledge about skyline queries, a short introduction of skyline queries consisting of illustrative examples is given. To provide non-experts in the audience with some background on distributed systems, different architectures of distributed systems focusing on highly distributed systems are also briefly presented. However, the detailed discussion of existing approaches will benefit also researchers with advanced knowledge in any of the above mentioned areas. To conclude, this tutorial does not require any knowledge on distributed skyline processing and aims at a broad range of researchers.

5. SHORT BIOGRAPHIES

Katja Hose is currently a post-doctoral researcher at the Max-Planck Institute for Informatics in Saarbrücken, Germany. She studied Computer Science at Ilmenau University of Technology, Germany and received her diploma in 2004. She joined the Databases & Information Systems Group at Ilmenau University of Technology as a research associate in the same year. She received her doctoral degree in Computer Science in 2009 and afterwards joined the Max-Planck Institute for Informatics in Saarbrücken. During her PhD studies she focused on distributed processing of skyline and top-k queries in schema-based P2P systems, efficient query routing, update strategies for routing indexes, heterogeneous data, and query rewriting using views. Her current research interests range from query processing and optimization in distributed systems, heterogeneous databases, and rank-aware query operators to information retrieval, linked data, and RDF query processing.

Akrivi Vlachou is currently a post-doctoral researcher at the Norwegian University of Science and Technology (NTNU) in collaboration with Athena Research and Innovation Center, Athens, Greece. She received her Ph.D. in 2008 from the Athens University of Economics and Business (AUEB), her MSc degree and her B.Sc. degree from the Department of Computer Science and Telecommunications of University of Athens in 2003 and 2001 respectively. In her dissertation, she studied methods for efficient query processing for highly distributed data. She has received fellowships for post-doctoral studies from European Research Consortium for Informatics and Mathematics (ERCIM) and from the Greek State Scholarship Foundation. She has published her research results in top-tier conferences and journals. Her research interests include query processing and data management in distributed systems, algorithms and query operators for large-scale data analysis and spatial-keyword search over web-accessible data.

6. REFERENCES

[1] S. Börzsönyi, D. Kossmann, and K. Stocker. The skyline operator. In *ICDE*, pages 421–432, 2001.

[2] L. Chen, B. Cui, and H. Lu. Constrained skyline query processing against distributed data sites. *TKDE*, 23(2):204–217, 2011.

[3] L. Chen, B. Cui, H. Lu, L. Xu, and Q. Xu. iSky: Efficient and progressive skyline computing in a structured P2P network. In *ICDCS*, pages 160–167, 2008.

[4] B. Cui, L. Chen, L. Xu, H. Lu, G. Song, and Q. Xu. Efficient skyline computation in structured peer-to-peer systems. *TKDE*, 21(7):1059–1072, 2009.

[5] B. Cui, H. Lu, Q. Xu, L. Chen, Y. Dai, and Y. Zhou. Parallel Distributed Processing of Constrained Skyline Queries by Filtering. In *ICDE*, pages 546–555, 2008.

[6] K. Fotiadou and E. Pitoura. BITPEER: continuous subspace skyline computation with distributed bitmap indexes. In *DaMaP*, pages 35–42, 2008.

[7] K. Hose, C. Lemke, and K. Sattler. Processing Relaxed Skylines in PDMS Using Distributed Data Summaries. In *CIKM*, pages 425–434, 2006.

[8] K. Hose, C. Lemke, K. Sattler, and D. Zinn. A relaxed but not necessarily constrained way from the top to the sky. In *CoopIS*, pages 339–407, 2007.

[9] K. Hose and A. Vlachou. A survey of skyline processing in highly distributed environments. *VLDB Journal*, pages 1–26. 10.1007/s00778-011-0246-6.

[10] Z. Huang, C. S. Jensen, H. Lu, and B. C. Ooi. Skyline queries against mobile lightweight devices in manets. In *ICDE*, page 66, 2006.

[11] H. T. Kung, F. Luccio, and F. P. Preparata. On finding the maxima of a set of vectors. *Journal of the ACM*, 22(4):469–476, 1975.

[12] H. Li, Q. Tan, and W. Lee. Efficient progressive processing of skyline queries in peer-to-peer systems. In *Infoscale*, page 26, 2006.

[13] D. Papadias, Y. Tao, G. Fu, and B. Seeger. An Optimal and Progressive Algorithm for Skyline Queries. In *SIGMOD*, pages 467–478, 2003.

[14] J. Pei, W. Jin, M. Ester, and Y. Tao. Catching the Best Views of Skyline: A Semantic Approach Based on Decisive Subspaces. In *VLDB*, pages 253–264, 2005.

[15] F. P. Preparata and M. I. Shamos. *Computational Geometry - An Introduction*. Springer, 1985.

[16] J. B. Rocha-Junior, A. Vlachou, C. Doulkeridis, and K. Nørvåg. AGiDS: A grid-based strategy for distributed skyline query processing. In *Globe*, pages 12–23, 2009.

[17] J. B. Rocha-Junior, A. Vlachou, C. Doulkeridis, and K. Nørvåg. Efficient execution plans for distributed skyline query processing. In *EDBT*, pages 271–282, 2011.

[18] A. Vlachou, C. Doulkeridis, Y. Kotidis, and M. Vazirgiannis. SKYPEER: Efficient subspace skyline computation over distributed data. In *ICDE*, pages 416–425, 2007.

[19] A. Vlachou, C. Doulkeridis, Y. Kotidis, and M. Vazirgiannis. Efficient routing of subspace skyline queries over highly distributed data. *TKDE*, 22(12):1694–1708, 2010.

[20] S. Wang, B. Ooi, A. Tung, and L. Xu. Efficient skyline query processing on peer-to-peer networks. In *ICDE*, pages 1126–1135, 2007.

[21] S. Wang, Q. H. Vu, B. C. Ooi, A. K. Tung, and L. Xu. Skyframe: a framework for skyline query processing in peer-to-peer systems. *VLDB Journal*, 18(1):345–362, 2009.

[22] P. Wu, C. Zhang, Y. Feng, B. Zhao, D. Agrawal, and A. Abbadi. Parallelizing Skyline Queries for Scalable Distribution. In *EDBT*, pages 112–130, 2006.

[23] Y. Yuan, X. Lin, Q. Liu, W. Wang, J. X. Yu, and Q. Zhang. Efficient computation of the skyline cube. In *VLDB*, pages 241–252, 2005.

[24] L. Zhu, Y. Tao, and S. Zhou. Distributed skyline retrieval with low bandwidth consumption. *TKDE*, 21(3):384–400, 2009.