International Journal of Database Theory and Application
Vol.8, No.5 (2015), pp.171-182
http://dx.doi.org/10.14257/ijdta.2015.8.5.14

Research Progress of Stream Data Query in Network Space

Yi Wu and Jianjun Zhou

School of Information Science and Technology, Heilongjiang University Harbin, Heilongjiang, 150080, China
wy51cn@aliyun.com, zhou_1969@tom.com

Abstract

In recent years, there has been widespread concern about the problems of stream data query both academic and industrial communities. The problems obtained some results. At the same time, big data stream brings great benefits for information society. Information query about stream data form has also brought crucial challenges. However, it is seldom about the research of big data stream query in network space. This paper analyzes the characteristics of stream data query in massive data, discusses the challenges and research issues of data stream for big data query. Finally the works for the data stream query are surveyed.

Keywords: *network space, big data, stream data query, concurrent processing architecture, quality analysis*

1. Introduction

With the rapid development of information technology, people gain unprecedented ability to collect and use data in networks, such as the Internet of things, social network, and mobile network. According to a research report of IDC, the amount of big data in network was 1.8ZB in 2011, and that amount is predicted to reach 35ZB by 2020, which means it will increase 50 times in the coming 10 years and the number of severs about managing data will get a 10-fold increase to cater for the 50 times growth of data. Features of big data is massive, multi-modal, to generate fast, high-value but low density, we cannot obtain information using traditional techniques and IT hardware and software tools within tolerated time.

At present, all kinds of stream data processing based on real time data has become the key point of the internet such as Internet and the internet of things. For instance, the amount of daily data processing in eBay platform is up to 100PB, exceeding the quantity in Nasdaq stock exchange. In order to analyze customers' shopping behaviours, eBay defines more than 500 types of data to track and analyze customers' behaviours [1]. Normally, big data in network which is in the form of stream generates dynamically and rapidly, with a strong timeliness. Data can be used effectively only if users have a good grasp of stream data. To cope with the last flowing big stream data like internet and sensor network, data processing systems must capture information keenly; then, it would create wealth for the information society. Big data applications for network, on the one hand is stream data at a high speed and on the other hand is a persistent historical data. It required comparing Stream data with historical data in real time and achieve personalized search. How to obtain the information based on fast flowing stream data becomes the new challenge of data query.

Compared with the traditional IT stream data query, it has distinct characteristics about big data stream query in network. This paper describe stream data processing systems in network space and related research work, analyze the characteristics for big data stream query, points out the objective about processing techniques for big data stream query, and

ISSN: 2005-4270 IJDTA

summarize the characteristics of query technology and faces challenge about big data stream management system.

In this paper, Section 2 discusses the characteristics, challenges and research issues of stream data query based on big data. Section 3 presents the stream data processing architecture in concurrent query model, stream data query technology and research methods. Section 4 also puts forward some expectation to the future work and makes this paper summary.

2. Characteristics, Challenges and Research Issues of Stream Data Query Based on Big Data in Network

2.1. Characteristics of Stream Data Query in Network

Apart from characteristics of conventional stream data query, there are some other characteristics about big data applications for network (Reference Figure 1). They are as followed:

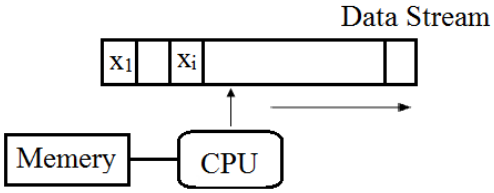

Figure 1. Basic Data Stream Model

(1) Instantaneity

Stream data is generated at high speed, so there must have a high time limit to the efficiency of data query. Comparing with traditional methods of stream data query, the methodology about big and high speed data which can meet the availability of results must be high real time.

(2) Usability

High speed and large scale stream data in network space can be considered infinite; thus, it cannot query after stored on hard disk or memory. So, data collection and generating results by one-time calculation are necessary. As prerequisites, obtained result sets are complete, which can meet the quality.

(3) Multi-source heterogeneity

Large scale stream data comes from heterogeneous internet and sensor network, which is consist of heterogeneous communication network, computing systems, storage systems and other physical devices.

(4) Uncertainty

The arriving of big scale dynamic stream data in network space is not influenced by external factors, and it changes with time and is affected by measurement errors, modal errors, environment noise and other uncertain elements. As a result, data query modal is in a passive position.

Considering about above characteristics, this paper discusses stream data query under the environment of large-scale, researches stream data parallel processing architecture technology, query mode, query model and query algorithm.

2.2. Challenges and Research Issues of Stream Data Query in Network Space

According to the analysis of the characteristics of stream data query, the research and implementation of the field faces many challenging. Taking into account the characteristics of the large number of big data in network, the fast generate speed, complex data type, the low value density, Main challenges and research issues are presented as followed:

2.2.1. Concurrent Processing Architecture of Stream Data Query

Based on conventional study of researches and technology, the research of real time stream data processing is divided into two types: centralized and distributed. In a centralized environment, stream data solutions only use hardware sources (generally refers to a single computer), with the limit of computing and storage sources, it can only deal with quite large data; cannot deal with large scale data. By contrast, in a distributed environment, aiming to dealing with data stream network which is composed of multiple processing operators, support data stream processing sources by calculating the scale of balancing multiple node operators. In this term, the processing ability is limited by single operator calculating ability, the cost of network communications and other sources.

At present, some scholars are exploring the combination of data stream and concurrent processing architecture (for example: MapReduce model of Hadoop architecture), for boosting the effectiveness of data processing [2]. And some other scholars are trying to use multicore processor chips for accelerating the process of data stream [3]. In generally, these researches are all in elementary stages.

2.2.2. Researches of Stream Data Query

Including traditional stream processing, technology and algorithms of data query pay more attention about the veracity and usability of algorithms, rather than handling large scale data sets, high dimensional data processing ability and the effectiveness of algorithms. Apart from them, there is not a high standard about the space and time complexity of algorithms.

With the development of information technology, problems about big data come out gradually, and the order of magnitude to deal with reaches to TB or even PB level. The growing tendency of big data would outweigh ability of processing data as well. In an environment of big data, fast flowing stream data has characteristics of unexpected and real time. Because the data volume is too big, even some data exists in distributed type, it is difficult to concentrate processing. Therefore, the calculation of big data is supposed to exchange from central, top-down model to decentralized, bottom-up and self-organization model [1].

To solve these issues, further researches about reaching linear and sub-linear algorithms are needed to ensure the availability of information query results and to meet the demand of big data query in the form of stream data in network space.

2.2.3 Corresponding Questions of Stream Data Query in a Big Data Environment

Researches about big data query in the form of stream data in network space, include stream data sliding windows query, approximate query technique, prediction query and skyline (preference) query (reference Table 1).

Table 1. The Literature Summary on Stream Data Query

No	The content of representative research		
	Topic	Methods	Reference numbers
1	Processing architecture	Query strategy	[4][5][6]
		Calculation model	[7]...[13]
		Hardware technology	[14][15][16]
2	Technology and method	Sliding window	[17]...[21]
		Approximate	[22][23]
		Compess	[24]
		Prediction	[25]...[28]
		Skyline	[29]...[45]
3	Quality analysis		[47]...[51]

With the fast development of internet, sensor network and other information technology, and to cater for the increasing social demands, stream data query will face tons of challenges. For instance, RFID (Radio Frequency Identification) is embedded into a variety of devices, and it acquires valuable information through allowing automatic recognition as well as acquiring stream data. As mobile phones, tablet computers and wireless navigations and other mobile devices are launched, mobile internet generates large scale stream data, which need new methodology of stream data query to meet the application of valuable information.

Besides, according to stream data morphology characteristics and business characteristics, data query can be researched in following aspects:

(1) One-time calculation efficiency of stream data. In order to improve stream data real time, to conquer the limit of resources such as storage space, transmission channels and processors, and to ensure the premise of results availability, we choose algorithms suitable for dealing with large scale data query (*e.g*. approximate calculation, compression algorithm).

(2) Privacy query in network space. In the network environment, some social organizations take privacy and business profits into considerations, they would handle heterogeneous multi-source large-scale data with independent safety mechanism and they can get data accessing authorization through Authentication or encryption. Hence, large scale stream data query with privacy protection is a challenge.

(3) Large scale stream data query in resource constraint fields. For example, in the sensor network field, due to energy constraints, to minimize excessive communication, more calculation is required. Or with the increasing bandwidth in internet, computing ability of multicore processing units strengthens, and memory wall phenomenon emerges. All these problems would lead to calculation bottleneck. Thus, data query algorithms with resource constraints (calculation, communication, storage, *etc*.) are needed to research.

3. Research Progress of Stream Data Query in a Large Scale Data Environment

Study on stream data at home and abroad for large data query has just started. Now we introduce research progress and tendency in next aspects:

3.1. Concurrent Processing Architecture of Stream Data Query

Differing from conventional processing architecture of stream data query, constructing a large scale one needs to integrate data from different channels, resources, and structures. It is also should emphasis factors of big data such as various types, fast flowing, dynamic mechanism and enormous value. Therefore, the parallel processing environment is an important foundation for large scale stream data processing.

3.1.1. The Stream Data Query Strategy in a Distributed Environment

Reference [4] proposes an adaptive scheduling strategy can be used in the distributed data stream processing environment. That can ensure response time and limited memory utilization. It is achieved by response time degrading gradually, and gets scheduling optimization. After scheduling completed, scheduling strategy would be changed based on current state of the system. Determine the global strategy through local sites. And the global task will be triggered only when local site cannot handle.

Reference [5] proposes a transited document GATES which can deal with distributed data stream on open network service architectures. It mainly aims at offering adaptive strategies based on constantly changed environment. According to various data stream environment, it can change corresponding sampling rates, approximate structure sizes and processing algorithms. For example, if data flowing speed accelerates, the system can acquire real time response by lower sampling rate. In order to support the self-adaptive, system designers need to provide transited document with parameters, these parameters allow the user to adjust the system in real time according to changes in the data stream environment. GATES sets up a simple performance model for predicting the relationship between the change of parameters and self-adaptive in and distributed environment.

Reference [6] proposes a loading shedding strategy which can query multi data streams. The process of reading data from the stream and extracting eigenvalue calculates with higher complexity, according to the utility of historical stream data items to determine whether to abandon the current stream data items. If current items eigenvalues are chosen not to exact, then they can be predicted based on historic data through low memory occupied Markov chains. The loading strategy can also be applied in distributed environment.

3.1.2. Efficient Parallel Stream Data Mode

Reference [7] introduces OpenMP, MPI, MapReduce and DryadLl and some other classical parallel programming models. OpenMP and MPI are in the relatively low level of abstraction models, which require programmers to explicitly deal with task management, data management and other details. Dryad tends to create a complete calculation process. Reference Figure 2, the Google MapReduce computing framework can provide adequate parallel computing semantic, focus on build calculation operators, and it also benefits large scale data processing division. Mapreduce is a popular classic model of processing concurrence structures.

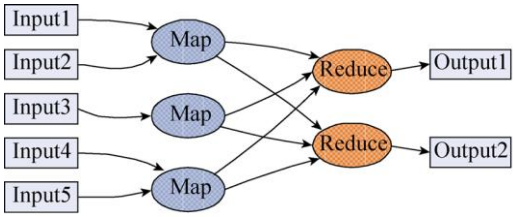

Figure 2. The Workflow of a MapReduce Program

Reference [8] describes a way to make Mapreduce suitable for stream processing and to achieve single-pass analysis scalable processing platform.

Reference [9] describes a C-MR system which can support stream processing persistent Mapreduce in 3 aspects. Firstly, extend the window concept of parallel stream processing to Mapreduce model. Sectionionondly, integrate various heterogeneous computing abilities. Finally, support flexible, dynamic workflow scheduling.

M3 system in reference [10] expands Mapreduce bypassing restrictions of HDFS, and achieves an effective stream processing system of full memory.

The characteristic of Stream Mapreduce event stream processing in reference [11] redefines Mapper and Reducer in Mapreduce and adds continuous, low-delay data processing ability.

Reference [12] regards fast flowing stream data as a representative of fast data processing in big data age. It considers original Mapreduce unsuitable for processing fast data. They design a similar framework: MapUpdate by combining characteristics of fast data. And they create a prototype system Muppet based on this framework.

Reference [13] combines the application of urban vehicles real time acquiring and processing, and then proposes a real time methodology which is mainly for processing large scale fast data stream. This methodology boosts some key technique bottlenecks such as local stage pipelines and intermediate results cache. Controlling the stage of pipelines based on system parameters makes the full use of CPU. It also optimizes local intermediate results high concurrent reading and writing performance by transforming memory and storage data structures, reading and writing strategies and replacing algorithms.

3.1.3. Hardware Technology of Stream Data Parallel Calculation

In reference [14], embed multi cores in a processor, and each core has computing ability. Calculation tasks are even divided into each core. Multicore Programming generally use multithreading development model. Currently there are two main decomposition models: task decomposition and data decomposition.

Reference [15] proposes a method to deal with frequent elements in a multi-core chip. This model based on cooperation mechanism is superior to the traditional design model based on competition mechanism.

Reference [16] proposes 2 calculating skyline methods which use multicore architecture; they are pskyline and parallel BBS.

3.1.4. Summary: There are tons of research results of stream query technique based on distributed processing architecture, but it is still hard to cater for the daily increasing large scale data processing. Hardware technology like multicore processors based on stream data parallel calculation is constrained by current technology level. Thus, there are few research results. With the development of cloud calculation, stream data processing concurrent processing frameworks of parallel programming models (Mapreduce is a good case) has started. Despite the lack of a comprehensive research and there are many issues remain unresolved, research in this area still has great potential.

3.2. Researches of Stream Data Query Technique and Methods

Domestic and international research work has been carried out for the stream data query, but with big data research network space has become a hot issue, stream data query researches based on big data are gradually being concerned about by vast number of researchers. Now we introduce research process and tend in terms of morphological characteristics and business characteristics.

3.2.1. Stream Data Sliding Windows Query

Reference [17] uses multi data stream concurrent technique to connect sliding windows. The paper proposes a cost model suitable for connecting data streams. Each connecting cost of algorithm is calculated by flow rate of each data stream. In other words, this model chooses low cost algorithms dynamically to connect sliding windows and query. Reference [18] gives a heuristic rule to select the appropriate connection order to reduce implementation costs.

Reference [19] proposes to achieve a connection processing operator which can deal with various input streams. In limited system resources, if data of connected sliding windows cannot be saved all, sample data would be saved. Then approximate query results of sliding windows can be revealed by sample data.

Reference [20] propose a compound sliding window model to compute the distinct values over basic sliding windows in an incremental way. The approach is well applicable for efficient decision making.

Reference [21] proposes a measuring approach to evaluate connecting approximation of sliding windows. That is connecting the number of results which is closer to the exact amount. The approximation is higher, and would come out with an optimization approximate sliding windows connection algorithms at the same time.

3.2.2. Approximate Query

Reference [22] proposes classic sampling approximation query technique. Using abstract common means of abstract data structures, acquire a little of sampling data as synopsis of data set from a multitude of data sets abiding by a proper rule. For example, accurate sampling and count sampling adopt set of <data, count> to keep record of different data and the number of appearance, rather than saving copies of the same figure. As this way goes, storage space is saved and more accurate approximate query is gotten. Sampling methods can be divided into uniform sampling and bias sampling. In uniform sampling, each element in the data set is selected in sampling set in the same probability. In bias sampling, different elements are selected in different probability, but the major item is how to determine the degree of sampling error.

Reference [23] proposes using wavelet decomposition at different levels on the function, to obtain a more accurate approximation query results. A user defines a parameter δ which is close to 0, it is possible to ensure error results in a small range because of the probability of $1-\delta$. The one-dimensional Haar wave extends to multi-dimensional Haar wave. A multi-dimensional data matrix after Harr wave decomposition of a multidimensional coefficient matrix, its space complexity is $O(B + \log N)$. Like one-dimensional Harr wave, multi-dimensional Harr wave gets original multi-dimensional matrix through coefficient matrix recovery.

3.2.3. Methods of Compression Query

Reference [24] suggests reducing memory demands of sliding windows through compression. It proposes SLZW and SALZW, as well as the query procedure of compressing data streams. Indeed, compression can diminish the storage space of sliding windows to some extent. However, when redundant information in the data stream is small does not fit to reduce its memory requirements through the data streams compression.

3.2.4. Prediction Query

Reference [25] proposes a method using multiple regressions to predict stream data, using mathematical statistical methods to establish relation function expressions between the dependent variables and the independent variables to predict the future development

trend of things. Then, the outputs of sensors are changed into analog linear stochastic systems, through studying historical trends sensor data to predict the value of which may be lost.

Reference [26] predicts algorithms based on rule matching. It proposes a methodology of predicting stream data by a model of generating frequent plot on an event sequence. At training stage, this method gets a model which describes characteristics of historic stream data. At prediction stage, according to the model, calculate the emerging probability of the most matching sub-sequence, and find a last non-overlapping occurrence which belongs to corresponding frequent plot. Then predict targets.

Reference [27] predicts algorithms based on rule matching of stream data. It adopts backward retrieving rule and antecedent strategy, using a complete binary tree with leaf nodes in a fixed number to store and maintain the data stream. In the reverse order of the rule antecedent topology, find the last antecedent in the smallest occurrence in order to achieve complete binary tree prediction. The required storage space is only about to the width of largest antecedent window in all the rules, rather than the amount of rules.

Reference [28] proposes to use an automatic machine for each general form of plot rule which are to be matched. Then trace the changes of automatic machines by singe-time scanning data stream. Hence, the last and the smallest antecedent non-overlapping occurrence can be hunted. In this term, Not only will the unbounded data stream is mapped to finite state space, and avoid quietly strict restrictions on the plot matching rules.

3.2.5. Skyline Multi-Objective Optimal Query

Skyline query has experienced the development process from centralized to distributed, from database to data stream, from certain data to uncertain data, from static data to dynamic data. In 2005, the first Skyline query algorithm on data stream was proposed by Lin and his group [29]. This algorithm queries Skyline set in a sliding window with the size N. With the enhancement of management technique of data steam, Skyline query on data stream has been brought out. Huang and others [30] took the lead in proposing the concept continuous Skyline under the background of mobile service. They also invented the model mixed static dimension and dynamic dimension, and generated the thought of continuous query.

Shengli Sun's group [31] first studied the definition of modeling and query based on sliding window aiming at probabilistic data stream, and optimized the system through the aspects of time and space. At the same time, Atallah [32] investigated the problem of Skyline query on discrete uncertain data sets. Zhang [33] studied processing Skyline query on uncertain stream based on tuple level uncertainty model.

Xin and his people [34] raised sliding window Skyline monitoring algorithms (SWSMA) to solve the problem of Skyline query of distributed data stream environment in sensor network. But it only fits sensor network. This algorithm cannot be used in high-speed distributed data stream environment. For this, Sun [35] investigated Skyline query problem in high-speed distributed data stream environment, focusing on reducing system response delay and communication load. They proposed a distributed method for solving non-sharing strategy asymptotic solution, and optimized key links of the method, which makes the method have better properties in communication load and response delay. But sources like bandwidth cannot be utilize efficiently, and the method doesn't fit in large scale networks.

Wu [36] first researched collateral progressive Skyline query problem on non-sharing architecture. He pipelined collateralized nodes which are involved in computation, and minimized the communication overhead between nodes. Mohammad [37] presents a method for selecting spatial objects. The system then computes a set of spatial objects in the preferred location considering the objects of the surrounding facilities by utilizing the idea of skyline queries. The approach is well applicable for efficient decision making. Cui

[38] reduced response time that restricts Skyline query procedure in large scale distributed environment by reinforcing the collateral processing between node groups. Aiming at collateral problems in multi-core processing architecture, Park [39] reduced comparison of dominance relationship between non-Skyline-query points by the order recombination of the access sequence of the data points, so as to decrease the I/O overhead and storage overhead. Yuan Wang's group [40] investigated fault-tolerant collateral processing Skyline query problem based on level division mode of data and found that better fault-tolerant processing capability can be obtain with lower storage overhead and communication overhead. In addition, according to the limitation of the energy, storage and handling capacity in wireless sensor networks, Haixiang Wang [41] had a comprehensive discussion of Skyline query method in wireless sensor networks. Currently, distributed collateral Skyline query basically aims at static data. But with the enhancement of Internet technology, development of collateral Skyline steam query is an inevitable trend.

Guangdong Wang and his group [42] developed the Skyline query algorithm on uncertain data steam, transferring centralized stream query to collateral processing. The problem that centralized algorithm lack of processing capacity can be solved through collateral execution. Lingli Li [43] proposed key words data steam query method based on Skyline, in order to improve XML data stream query. With the development of cloud computing technology, such high parallel processing frameworks as MapReduce model sharply improve the efficiency of Skyline query process. Reference [44] raised a collateral algorithm to deal with Skyline query under MapReduce model, but researches in Skyline query of mass steam data have not been carried out so far. There are a lot of questions waiting for scientists to explore. For example, the problem of Skyline query under the communication restriction in distributed concurrent environment. Even though Sun [45] proposed Skyline query algorithm in bandwidth constrained distributed environment, FDS, and Jin Huang [46] studied Skyline query problem in high speed bandwidth environment. In recent years, more and more distributed collateral computing environment with high capacity has been disposed. Along with that is high speed communication bandwidth. It brings unprecedented opportunities for analyzing mass data stream in cyberspace, as well as new technical challenges.

3.2.6. Summary

Researches on big scale data query in network space are in elementary stages, there is little achievement in compression query. At present, mainly research processing distributed query, approximate query and prediction query combining concurrent algorithms. In next step, combining characteristics of stream data, further researches about effective query with linear and sub-linear calculation complexity are needed to improve query efficiency.

3.3. Quality Analysis Methods of Stream Data Query

Reference [47] studies how to acquire stream data in sensor network in the premise of maintaining availability of data. It comes up with a stream data frequency acquisition algorithm by adopting Hermit interpolation and cubic spline interpolation. This method can get maximum data collection with minimum acquisition, and maintain the quality of stream data.

Due to the existence of data redundancies in sensing nodes of geographical proximity, reference [48] proposes stream data acquisition method that is location-sensitive. It utilizes geographical characteristics of data resources to filter redundant data. Then it enhances data quality in the event detection application and reduces probability of miscarriage of justice.

Reference [49] proposes a minimum quality calculation rule based on data item groups and the credibility, mining algorithms and an idea of detecting error data through quality criteria. The data quality criteria imitate credibility evaluation mechanism of association rules and expressive ability of condition functions, as well as uniformity describe functional dependencies, condition function dependencies and association rules. The criteria have characteristics of simple, objective, comprehensive, and accurate detection of abnormal data.

In reference [50], provenance reveals the whole procedure of data generation and changes with time. That is for data quality assessment, data check, recovery and quotation. The data provenance can be divided into different data evolution process and the same data source evolution process, namely the schema level and instance level data evolution. The performance and query of schema level and instance level data provenance as main line to describe the research procedure of provenance. Model level provenance introduces provenance tracking technology of query rewrite and schema mapping. Instance level provenance summaries recent research procedure in aspects of relational data, XML data, stream data.

Reference [51] introduces basic concepts of big data availability, discusses challenges of big data availability, and probes into problems of big data availability, and summaries some results. And it proposes big data availability theoretical system, acquisition theory and means, errors automatically finding and automatically fixing algorithms, approximate calculation theory as well as mining theory and algorithms of weak available big data.

Overall, academia conducts a multitude of research about stream data, but there are little achievements in quality of big data query in network and analysis methods of availability of streaming query. Nevertheless, the demand of data query on valuable data in application field is quite urgent. So there are still numerous problems need to be solved in query quality analysis.

4. Conclusions

Researches about stream data query based on big data in network space are in preliminary stages. Many researches stem from traditional distributed query, concurrent stream data query. In recent year, with the development of theory and applications of dynamic, multi-dimensional, complex big data, these characteristics make conventional approaches of data analysis and data processing not suit any more. For example, high speed rate, large volume stream data exists in distributed form, which is hard to process in concentrated way. Streaming query results of big data complicated types, semi-structured, unstructured and other non-relational data are insufficient. Hence, we need to carry out data stream characteristics and business characteristics research of data query systematically, to discuss the characteristics of network stream data with large data query model, query, query technology to be directed against stream data concurrent processing construct technique, methods and technology of stream data query stream data characteristics, the business of stream data query. Then discuss the models, methods and technology of stream data query with the characteristics of big data in network space. In conclusion, in spite of the absence of comprehensive researches about stream data query based on big data, application prospect in this field is decent, so the research work is of great value and academic potential.

Acknowledgment

This work is supported by the Nature Science Foundation of Heilongjiang Province with the grant number: F201325. The author would like to thank for their support.

References

[1] W. Y. Zhuo, J. X. Long and C. X. Qi, "Network Big Data: Present and Future", Chinese Journal of Computers, vol. 36, no. 6, **(2013)**, pp. 1125-1138.
[2] Li B., Mazur E., Dian Y., McGregor A. and Shenoy P., "A platform for scalable one-pass analytics using MapReduce", Proceedings of the 2011 ACM SIGMOD International Conference oil Management of Data. Athens, Greece, **(2011)**, pp. 985-996.
[3] Das S., Antony S., Agrawal D. and Abbadi A. E, "Thread cooperation in multicore architectures for frequency counting over multiple data streams", Proceedings of the VLDB Endowment, Z, no. 1, **(2009)**, pp. 217-228.
[4] Ghoting A. and Parthasarathy S., "Facilitating interactive distributed data stream processing and mining", Proceedings of the International Parallel and Distributed Processing Symposium (IPDPS), **(2004)**.
[5] L. Chen, Reddy K. and Agrawal G., "GATES: A grid-based middleware for processing distributed data streams", Proceedings of the International Symposium on High Performance Distributed Computing (HPDC), **(2004)**, pp. 192-201.
[6] Chi Y., Yu P. and Wang H., "Loadstar: A load shedding scheme for classifying data streams", Proceedings of the SIAM International Conference on Data Mining (SDM), **(2005)**, pp. 342-361.
[7] Isard M., Budiu M., Yu Y., Birrell A. and Fetterly D., "Dryad: Distributed data parallel programs from sequential building blocks", Proceedings of the 2007 Eurosys Conference (Eurosys 2007), Lisbon, Portugal, **(2007)**, pp. 59-72.
[8] L. Boduo, Mazur E. and D. Yanlei, "A platform for scalable one-pass analytics using MapReduce", Proceeding of SIGMOD 2011, New York: ACM, **(2011)**, pp. 985-996.
[9] Backman N., Pattabiraman K. and FonSectioniona R., "C-MR: Continuously executing MapReduce workflows on multi-coreprocessors", Proceeding of the 3rd Int. Workshop on MapReduce and Its Applications. New York: ACM, **(2012)**, pp. 1-8.
[10] Aly A. M., Sallam A. and Gnanasekaran B. M., "M3: Stream processing on main-memory MapReduce", Proceeding of ICDE 2012.Piscataway, NJ: IEEE, **(2012)**, pp. 1253-1256.
[11] Brito A., Martin A. and Knauth T., "Scalable and low latency data processing with stream MapReduce", Proceeding of CloudCom 2011. Piscataway, NJ: IEEE, **(2011)**, pp. 8-58.
[12] W. Lam, L. Lu and Prasad S., "Muppet: MapReduce style processing of fast data", PVLDB, vol. 5, no. 12, **(2012)**, pp. 1814-1825.
[13] Q. K. Yuanl, Z. Z. Fen, F. Junl and M. Qian, "Real-Time Processing for High Speed Data Stream over Large Scale Data", Chinese journal of computers, vol. 35, no. 3, **(2012)**, pp. 477-490
[14] G. X. Qin, J. C. Qin, W. X. Lin, Z. Rong and Z. A. Yin, "Data-Intensive Science and Engineering: Requirements and Challenges", Chinese journal of computers, vol. 35, no. 8, **(2012)**, pp. 2563-1578.
[15] Das S., Antony S., Agrawal D. and Abbadi A. E., "Thread cooperation in multicore architectures for frequency counting over multiple data streams", Proceedings of the VLDB Endowment, Z, no. 1, **(2009)**, pp. 217-228.
[16] Park S., Kim T., Park J., Kim J. and Im H., "Parallel skyline computation on multicore architect res", Proceedings of the 25th International Conference on Data Engineering. Shanghai, China, **(2009)**, pp. 760-771.
[17] J. Kang, J. F. Naughton and S. D. Viglas, "Evaluating Window Joins over Unbounded Streams", ICDE Conference, **(2003)**.
[18] L. Golab and M. Tamer, "OZSU. Processing sliding window multi-joins in Continuous queries over data streflms", Waterloo University Technical ReDOrt CS-2003-01.February, **(2003)**.
[19] Stratis D., Viglas J. F. and N. Jo. Burger, "Maximizing the Output Rate of Multi—Wav Join Queries over Streaming Information Sources", In Proceeding of the Intl. Conf. on Very Large Data Base, **(2003)**.
[20] Y. Zhong, J. Zhu, M. Ren and Y. Yang, "Estimation of the Number of Distinct Values over Data Stream Based on Compound Sliding Window", Journal Of Software, vol. 8, no. 1, **(2013)**, pp. 19-24.
[21] A. Das, J. Gehrke and M. Riedewald, "Approximate Join Processing Over Data Streams In Proceeding of the ACM SIGMOD Intl. Conf. on Management of Data, **(2003)**.
[22] B. Babcock, M. Datar and R. Motwani, "Sampling from a Moving Window over Streaming data In Proceeding of SODA 2002, January **(2002)**.
[23] Gilbert, Y. Kodidis and M. Strauss, "Surfing wavelet on streams: One-pass summaries for approximate aggregate queries. In Proceeding of VLDB, **(2001)**.
[24] W. Xu, L. J. Zhong and W. W. Ping, "Processing Compressed Sliding Window Continuous Queries over Data Streams", Journal of computer research and development, vol. 41, no. 10, **(2004)**, pp. 1639-1644.
[25] Fletcher A. K., Rangan S. and Goyal V. K., "Estimation from lossy sensor data: Jump linear modeling and Kalman filtering", in: Proceeding of the 3rd Int'l Symp. on Information Processing in Sensor Networks. **(2004)**, pp. 251-258.
[26] Laxman S., Tankasali V. and White R. W., "Stream prediction using a generative model based on frequent episodes in event sequences", In: Proceeding of the 14th ACM SIGKDD Int'l Conf. on Knowledge Discovery in Data Mining. New York: ACM Press, **(2008)**, pp. 453-461.

[27] Cho C. W., Zheng Y., Wu Y. H. and Chen A. L. P., "A tree-based approach for event prediction using episode rules over event streams", In: Proc. of the 19th Int'l Conf. on Database and Expert Systems Applications, **(2008)**, pp. 225-240.
[28] Z. H. Sheng, W. Wei and S. B. Le, "Data Stream Prediction Based on Episode Rule Matching", Journal of Software, vol. 23, no. 5, **(2012)**, pp. 1183-1194.
[29] X. Lin, Y. Yuan and W. Wang, "Stabbing the sky: Efficient skyline computation over sliding windows", In: Proceedings of the 21st International Conference on Data Engineering (ICDE2005), Tokyo, Japan, April, **(2005)**, pp. 502-513.
[30] Huang Z., Lu H. and Ooi B., "Continuous skyline queries for moving objects", IEEE Transactions on Knowledge and Data Engineering, vol. 18, **(2006)**, pp. 1645-1658.
[31] S. S. Li, D. D. Bo, H. Z. Hua, Z. Q. Xun and Z. L. Xin, "Algorithm on Computing Skyline over Probabilistic Data Stream", Acta Electronica Sinica, vol. 37, no. 2, **(2009)**, pp. 285-293.
[32] Atallah M. J. and Qi Y., "Computing all skyline probabilities for uncertain data", Proceedings of the twenty-eighth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems. ACM, **(2009)**, pp. 279-287.
[33] Zhang W. J., Lin X. M. and Zhang Y., "Probabilistic Skyline Operator over Sliding Windows", In: ICDE, **(2009)**.
[34] Xin J., Wang G. and Chen L., "Continuously maintaining sliding window skylines in a sensor network", In: the 12th International Conference on Database Systems for Advanced Applications. Berlin: Springer-Verlag, **(2007)**, pp. 509-521.
[35] S. S. Li, L. J. Jiu and Z. Y. Yong, "Efficient Processing of Continuous Skyline Query over Distributed Data Streams", Journal of Software, vol. 20, no. 7, **(2009)**, pp. 1839-1853.
[36] Wu P., Zhang C. and Feng Y., "Parallelizing skyline queries for scalable distribution", In: EDBT, Munich, Germany, **(2006)**, pp. 112-130.
[37] M. S. Arefin, X. Jinhao, C. Zhiming and Y. Morimoto, "Skyline Query for Selecting Spatial Objects by Utilizing Surrounding Objects", Journal of Computers, vol. 8, no. 6, **(2013)**, pp. 1742-1749.
[38] Cui B., Lu H. and Xu Q., "Parallel distributed processing of constrained skyline queries by filtering", In: ICDE, **(2008)**, pp. 546-555.
[39] Park S., Kim T. and Park J., "Parallel skyline computation on multicore architectures", In: ICDE, **(2009)**.
[40] W. Yuan, W. Yijie, D. Ruipeng and P. Xiaoqiang, "Fault-Tolerant Parallel Skyline Computation in Cloud Computing Environment", Journal of Frontiers of Computer Science and Technology, vol. 5, no. 9, **(2011)**, pp. 804-814.
[41] W. H. Xiang, Z. J. Ping and S. B. Li, "Skyline Query Processing in Wireless Sensor Networks", Computer Science, vol. 40, no. 8, **(2013)**, pp. 14-23.
[42] W. Guangdong, W. Yijie, L. Xiaoyong and W. Yuan, "Parallel Skyline Computation over Uncertain Data Streams", Journal of Frontiers of Computer Science and Technology, vol. 6, no. 12, **(2012)**, pp. 1116-1125.
[43] L. L. Li, W. H. Zhi, G. Hong and L. J. Zhong, "Efficient Top-K Keyword Search on XML Streams", Journal of Software, vol. 23, no. 6, **(2012)**, pp. 1561-1577.
[44] D. Lin, L. Xin, J. C. Wan, G. G. Ren, H. Shan, "Efficient Skyline Query Processing of Massive Data Based on Map-Reduce", Chinese Journal of Computers, vol. 10, no. 34, **(2011)**, pp. 1785-1796.
[45] Zhu L., Tao Y. and Zhou S. G., "Distributed skyline retrieval with low bandwidth consumption", IEEE Transactions on Knowledge and Data Engineering, vol. 21, no. 3, **(2009)**, pp. 384-400.
[46] Huang J., Zhao F. and Chen J., "Towards Progressive and Load Balancing Distributed Computation: A Case Study on Skyline Analysis", Journal of Computer Science and Technology, vol. 25, no. 3, **(2010)**, pp. 431-443.
[47] C. Siyao and L. J. Zhong, "O(ε)-Approximation to Physical World by Sensor Networks", Proc. of IEEE INFORCOM'13. Piscataway, NJ: IEEE, **(2013)**, pp. 3184-3192.
[48] C. Siyao, L. J. Zhong and L. Yu, "Location aware peak value queries Sensor Networks", Proc of IEEE INFORCOM'12. Piscataway, NJ: IEEE, **(2012)**, pp. 486-494.
[49] L. Bo and G. Y. Rong, "Mining Method for Data Quality Detection Rules", PR & AI, vol. 25, no. 5, **(2012)**, pp. 835-844.
[50] G. Ming, J. C. Qing, W. X. Ling, T. X. Xia and Z. A. Ying, "A Survey on Management of Data Provenance", Chinese Journal of Computers, vol. 33, no. 3, **(2010)**, pp. 373-388.
[51] L. J. Zhong and L. Xianmin, "An Important Aspect of Big Data: Data Usability", Journal of Computer Research and Devolopment, vol. 50, no. 6, **(2013)**, pp. 1147-1162.

Author

Wu Yi, was born in 1963, professor. His current research interests include large scale data processing, data mining, sensor network, *etc.*

