International Journal of Hybrid Information Technology
Vol.7, No.3 (2014), pp.177-186
http://dx.doi.org/10.14257/ijhit.2014.7.3.18

A Framework for Human Resource Information Systems Based on Data Streams

Wei Chen

Department of Human Resources, Chongqing University of Posts and Telecommunications, Chongqing 400065, China
E-mail: cweicqupt@163.com

Abstract

One of the key issues for a successful business is human resource management and that process is under significant influence of modern information technology. Human Resources Information Systems (HRIS) are systems used to collect, record, and store, analyze and retrieve data concerning an organization's human resources. The study of dynamic data management is one of the burning topics among the HRIS. Although it has been studied extensively, a new kind of data, named streaming data, which have been appeared on the HRIS such as real-time monitoring of employee, is not well understood, because of its rapid data arriving speed and huge size of data set. This paper describes in the basic model and the architecture of a new system to manage human resource data streams for real-time applications. We first provide an overview of the elementary model and architecture and then describe in detail a stream-oriented set of operators. Ultimately, the experimental result shows that the model offers a better way of dynamic human resource data management that cannot be achieved with existing schemes.

Keywords: *Human Resources Management System (HRMS), Information Systems, Data stream management, Continuous query*

1. Introduction

Social and organizational changes in the economic environment are numerous and extensive. According to that, it is important for human resource management (HRM) to be comprehensive, high in quality, fast, flexible and in line with upcoming trends, because it is one of the parameters of successful business [1]. Use of information and communication technology becomes an imperative for HRM, as well as the other activities in the business [2].

Growth of Human Resources Management System (HRMS) continues, and the HRMS function consists of tracking existing employee data which traditionally include personal histories, skills, capabilities, accomplishments, etc. The purpose of any HRIS is to provide its users with information. For the most part, a HRIS uses a relational database to store and categorize all the separate data files, which are linked by some common elements. The implementation of a new technology is often accompanied by a dramatic change in human resource (HR) management policy. According to that, effective HRM, in order to provide competitive advantages, requires adequate updated information. IT evolution has improved a technique of collecting this information through the development of HRIS systems.

A problem in HRMS surfaces when the applications scale in terms of number of data providers and data consumers and in the richness of information exchanged. At the same time, the study of dynamic data plays an increasingly important role in database research, there are lots of data stream management system (data stream management system, referred as DSMS), including the STREAM project at Stanford University [3], the

Telegraph project of the University of California [4], Berkeley, Brown University, and hemp Institute of Technology cooperation Aurora project [5]. Industry-specific backgrounds, more comprehensive data management solution are given. On the other hand, data mining technology based on flow data model has been widely studied, including doing cluster analysis, decision tree analysis and density estimation [6, 7]. The core of the data stream is the summary of the design of data structures. Traditional solution methods have suffered from the classical ensemble average limitation presented by analysis of low-level characteristics, therefore, the human resource data gathered are sometimes inconclusive and, in part, contradictory. Different base classifiers can be incorporated into this framework according to diverse application requirements, which provide the flexibility of using this approach across a wide range of domains.

To overcome these problems, we will begin by looking at some defensible models, which are suitable for management of dynamic human resource data. Streaming model provides a multifaceted, dynamic, and robust framework for assembling real-time applications, which is appropriate for our aim. In this paper, we propose a general framework that is able to stream data. Dynamic message can be incorporated into this framework according to different application requirements, which provide the flexibility of utilizing this approach across a wide range of domains.

2. Related Works

2.1. Real-Time Data Management

For the first several decades of database research and development, the dominant paradigm was that database management systems (DBMSs) stored databases primarily on magnetic disks. The DBMS's job was to process a stream of updates and queries against such disk-resident databases.

In the 1980s and early 1990s, several data-base research and development efforts focused on the main memory (RAM) based DBMSs built primarily to address very high transaction processing loads or other real-time applications. As high-speed networks (such as the Internet) began to emerge, researchers (and then developers) started to deal with distributed data management issues, addressing topics such as distributed transaction management and query optimization in distributed settings (minimizing query processing's communication costs, for example)

As networks have become faster and more widespread, the use of data stream management has risen. The growing use of sensor networks has also stimulated research in this area. In both cases, the goal is to process data as it streams from the communications network in real (or nearly real) time. Such processing occurs largely in core memory, using minimal magnetic disks, which are usually seen as either too slow or too energy-intensive to be usable with high-bandwidth communication networks or bandwidth and power constrained sensor networks. Other major influences on data stream management have been event notification systems and publish–subscribe systems. Event notification systems disseminate event notices across computer networks, combine simple event notices into complex event notices, and selectively disseminate such notices to interested parties. These systems are widely used in process control and network monitoring systems.

In addition, publish subscribe systems also deal with disseminating messages across computer net-works and the selective "subscription" to such messages via individual processes. Content-based routing or subscriptions are in force continuous queries on publication streams. These systems are closely linked to data stream management, but lack notions of aggregate queries, for example. Widespread applications include financial news dissemination. Data stream management thus represents the confluence of ideas from many areas of database management research.

According to that, data stream management thus represents the confluence of ideas from many areas of database management research.

2.2. Data Stream Management

Data stream management presents several strategic challenges. Firstly, it must represent boundless streams of data in finite memory, either via moving averages, exponential damped estimators, sliding windows, random sampling, or hidden Markov models (HMMs). This is essential to reconcile the infinite-length input data streams and the finite memory resources data stream management systems must cope with.

Additionally, data stream management systems are frequently subjected to rigorous resource constraints in part, because they operate primarily in main memory. In other settings, as with sensor networks, such systems might have energy availability or communications bandwidth constraints. Thus, data stream management must address resource-constrained (RAM, energy, or communications bandwidth) query processing (including aggregate queries), and optimization.

Because many data stream management applications are concerned with state estimation of distributed real-time systems (such as electric power grids), we must pay careful attention to temporal issues especially time stamps, time synchronization, and minimization of temporal skewers to assure that state estimates are consistent.

Data stream management systems usually can't afford to either store or reprocess an entire input data stream, so they must take decisions and computations as the data arrive. Novel online query processing algorithms help solve this problem.

Much previous work has sought to optimize latency and memory on massive data streams, but Theodore Johnson and his colleagues first discussed the idle-waiting problem caused by mismatched arrival rates of input streams in binary operators [8]. Their solution uses punctuation tupelos, which have proven useful in dealing with problems such as blocking operators [9]. Data stream joins [10], and out-of-order tuples [11]. A related approach is Aurora's time-out mechanism [12], in which tupelos are discarded if they wait for future tuple arrivals for longer than a definite time threshold. The time-out way resolves the idle-waiting problem but sacrifices the correctness of query results. It might therefore be pleasing for the Aurora aggregate operators [13], but much less so for operators such as unions and joins. Bai and his colleagues first discussed the use of on-demand time stamps for idle waiting prone (IWP) operators [14], but didn't cover unary operators, brashness, and the results of other experiments we've discussed here. Others have discussed related techniques to support composite event semantics [14] and RFID applications [15, 16].

2.3. Human Resource Information Management

In the last two decades researchers have started to show interest in the field of HRIS though they focused more on areas such as predominate of HRIS [17], conditions for successful usages [18], use of HRIS and current usages patterns [19], areas in HRIS implementation [20, 21, 22], and achieving competitive advantage [23]. Current studies have investigated HRIS adoption determinants in Singapore and Australia [24, 25]. Van Vo proposes a novel framework based on data mining technologies for making a prediction of business environment [26]. However, these authors agreed upon there is a paucity of research in the area and especially it is necessary to investigate to which extent those factors affect adoption of the system. Further work is also essential in addressing HRIS adoption in the private sector organization as research is currently lacking in those areas. Thus, researchers aim to investigate influencing factors of HRIS adoption, identify to what extent those factors affect the HRIS adoption and finally, examine the relationship between factors influencing the adoption of HRIS and perceived effectiveness of HRIS.

There are lot of terms in use for these systems, but the most common is the following: e-HRM (e-Human Resource Management), HRIS (Human Resource Information Systems) and HRMS (Human Resource Management Systems). We must know that there is an elemental difference between e-HRM and HRIS. HRIS, as human resource information system, has straightforward implementation in the HR department and employees in this department are users of that system. Enhancement of HR Department is the principal goal of HRIS, which will indirectly improve business. Term e-HRM covers services not only for HR department, but also for a wider range of employees, potential employees and management. Those services are available over Internet or Intranet. The difference between HRIS and e-HRM could be described as a transition from the automation of HR Services (Transactional Systems) to IT support of HR information (Management Information Systems). Apart from that, HRIS can be seen as a database system or a series of interconnected databases, and HRMS as software that can combine multiple HR functions [27]. The differences between the two systems are quite blurred so these terms are in use as synonyms in many references. In this paper term HRIS will be used in a wider context, because the aim of this paper is not intended to define the term precisely, but to give an insight into the importance and necessity of introduction of modern information technology in the processes of HRM. The aim is to comprehensively show the role of HRIS systems, their evolution, structure, advantages and possible shortcomings, as well as the process of implementing systems in the organization and to highlight the importance they play in modern business.

However, this particular paper only focuses on a part of the whole research and aims to identify factors influencing HRIS adoption in organizations. This investigation was done using archival research method where the researchers analyzed the data based on secondary resource.

2.4. Research challenges

Research challenges in designing a storage framework for Human Resource data includes the following.

1. Support for a scale out architecture. An extensible architecture that can assimilate nodes one at a time to assist increased HR data storage requirements.

2. High throughput storage of data. Given the number of data sources, the framework must be able to store data streams arriving at high rates.

3. Efficient retrievals of specific portions of the data. Given the large data volumes involved we must support fast sifting of stored data streams in response to queries that target a distinctive feature at a specific time for a given use.

4. Fast detection of queries. Often the query parameters are adjusted based on results from past queries. To support fine tuning of queries, the framework must have accurate and efficient detection of situations when there are no data that match a specified query.

3. Problem Definition

Dynamic human resource data are assumed to come from a variety of data sources such as Internet of Things. We will use the term data source for either case. In addition, a data stream is the term we will utilize for the collection of data values presented by a data source. Each data source is assumed to have a matchless source identifier. The architecture of data streams is shown as Figure 1.

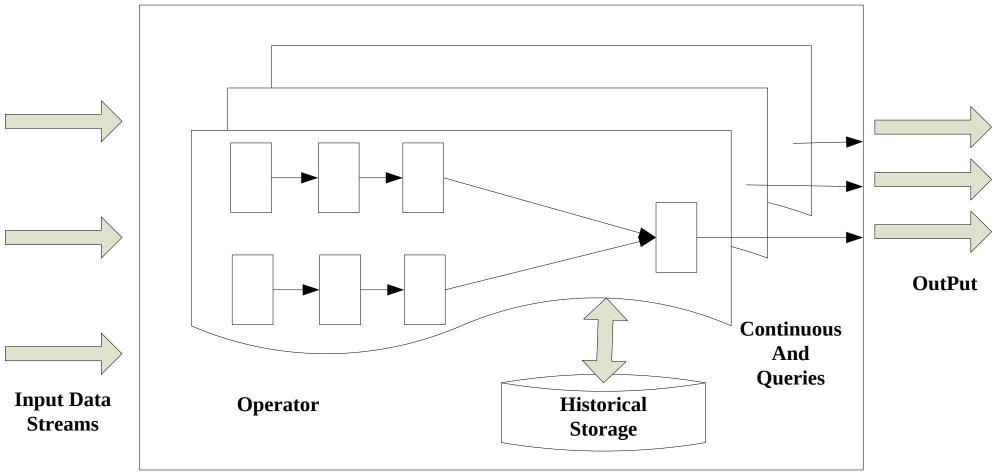

Figure 1. Architecture of Data Streams

Usually, A DSM system (DSMS) is defined as the following 4 tuple:

DSMS= 〈*H,OT,SPD,SP*〉 DSMS= 〈*H,OT,SPD,SP*〉 where H is a finite set of hosts participating in the DSM system, OT the finite set of globally available operator types offered by all hosts, SPD is a set of available stream process definitions, and SP is the set of running stream processes.

We make a set of definitions as blow:

Definition 1. A block is a array of data related to the monitor of HR, We define a new block as data set D_{t-1}, with m instances: $\{X_i, Y_i\}$, (i = 1, …, m), where X_i is an instance in the feature space X and $Y_i \in Y = \{1, 2, . . . , c\}$ is the class identity label associated with X_i.

Definition 2. We define the combination stream A, to be a stream that represents the combined stream of all participants, where the combined stream is the sum of the participant streams. In addition, a new data set D_t, with m' instances, where m' may or may not be the same size as m, and can be represented as $\{x_j , y_j \}$, (j = 1, . . . ,m').

In this context, the number of items in B is defined as j, i. It represents the width of the burst and j, i⩽n, the size of each window. Also, observe that i and j can be equal, so a burst can contain a single item.

Definition 3. We define B as a subset of a stream A, which having a magnitude that deviates significantly from the average magnitude of previous windows of a stream.

In the simplest case, creating the combined stream requires communicating each participant's stream to a central location. The time needed to send a participant's stream is the participant's communication cost.

Definition 4. Uncertain Data Stream (UDS). An uncertain data stream (UDS) contains a sequence of uncertain contents, formally, UDS = $\{S_1, S_2, \dots , S_N\}$, S are in increasing order of their arrival time, and we assume the element S_i arrives at timestamp i. Each uncertain tuple S_i is only alive between its arriving and expired timestamps, which is determined by the size.

Based on the above definitions, Figure 2 shows the scheduling mechanism. We can see that data transfer between tasks carried out by the group of business objects, business objects, the business group is a container that holds the object, which is the exchange of data between tasks pipeline. Tasks connected with the business object group by binding up on binding definition of the task group objects to the business object access rule.

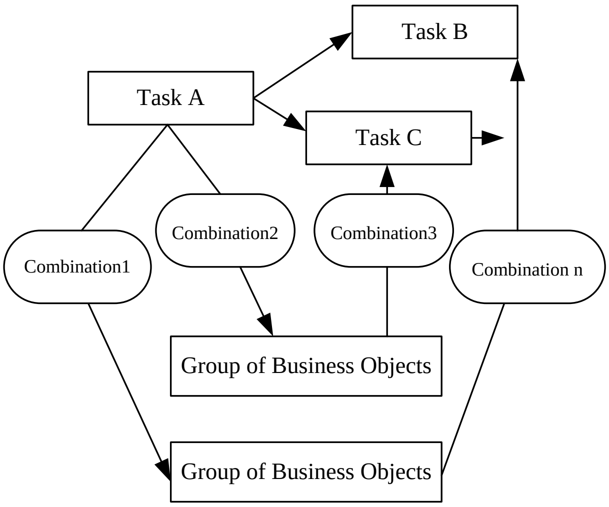

Figure 2. The Schematic diagram for scheduling mechanism

4. Key Methods

4.1. Data Blocks Management

Data blocks enter the system as a stream of bytes. Once the data is received, the metadata portion of the block is desterilized so it can be read and indexed in the in the memory, and then the data is written to disk.

Our storage scheme involves creating several file system level objects. As a block enters the system, its metadata and content are stored separately on disk, which creates two files and therefore consumes two nodes on the file system. In addition, directories are also created for the on-disk hierarchy that resembles the memory graph. The DSMS I/O structure is shown as Figure 3.

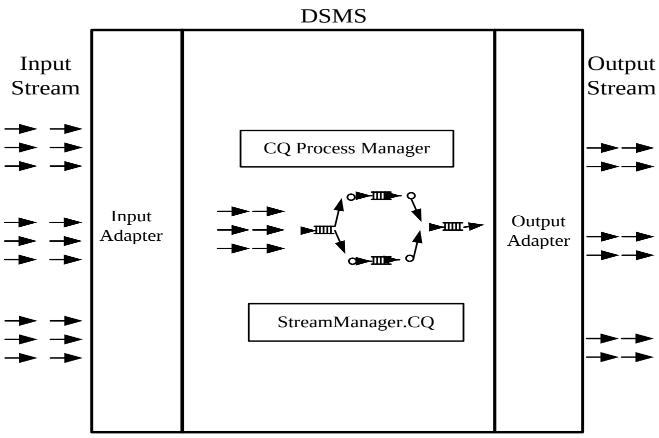

Figure 3. DSMS Structure

4.2. Failure Recovery

In the event of power loss, system crash, or scheduled reboot, etc. The framework must recover the state after restarted. Firstly, the recovery involves reading the system map, which contains enough information edge to restore the system graph that is utilized to create datasets.

In addition to that, we describe a failure recovery method based on the given HRMS model and analyze the consequences of failures of the operator, data stream and host levels, respectively. Note that this failure model encompasses the failures of all operator instances on a particular host, but also takes into account that individual operator instances at a host might fail, for instance due to an 'out-of-memory' exception in an overload situation, without impacting other instances at that host. The latter situation is such as to occur, for example, if operator instances are hosted on resource-limited (mobile) devices.

It should be recognized that the approach presented in this paper jointly addresses different types of failures, including failures at operator level. This is in contrast to other work in the field, such as [28], that only focuses on failure handling for host granularity. However, more fine-grained failure handling at operator level is beneficial for two principal reasons. Firstly, the approach of this paper is tailored to handle failures in mobile environments which are characterized by devices with limited resources. In such environments, operator failures may be triggered by overload situations. Typically, memory or CPU load limits is reached and some running operator instances on a mobile host may suffer from memory allocation errors or thread starvation. This leads to the situation that one or more operators on a mobile host are subject to operator failures. Addressing failure handling at operator level allows restricting operator migration to these failed operators while keeping the additional operator instances that are working properly on the device. Secondly, the more fine-grained approach at operator level subsumes failure handling at the node level. However, it allows to individually treating all local operators that fail due to a node failure. This means that they can be migrated independently of each other.

4.3. Rule Description

Queries in the framework are technically more than just a query. They are actually If-Then rules, where the IF clause contains the SQL query and the THEN clause contains an action, a user-defined function that is executed when the result of the query is a non-null set. The combined construct leverages the strengths of SQL queries for filtering and limited aggregation (*i.e.*, sum, max, and min) and the action for manipulation of the data set

As an example, the rule R:1 shown below, accepts three event types, all listed in the FROM clause: data events from the HR model as Data_Hr a user request as Request_Hr, and a performance monitoring event as Perf_ Hr. The condition in the WHERE clause evaluates to true if the data event from the model is at least partially contained in the region bounded by the min and max time provided by the user and the current network is better than a threshold specified by the constant MAX_H If the query is satisfied, the function "ppk2ppm" is triggered and passes the events that satisfied the query (one of each event type). The function "ppk2ppm" converts species concentrations from parts-per- kilometer to parts-per- million. Generally the employee monitor data is stored in database at first, and then users obtain data from database to judge the traffic status. The database carries a great burden because of the great amount of data, and it also presents the problem that a real-time assessment cannot be reached. This is neither effective nor reasonable.

```
CREATE RULE R:1 ON Data_ Hr, Request_ Hr, Perf_ Hr
IF
SELECT Data_ Hr
FROM Data_ Hr as d, Request_ Hr as r,
Perf_ Hr as p
WHERE
(d.tim_min > = r.tim_min
p.latency < = MAX_H and p.aid = r.aid
```


THEN
FUNC ppm2ppb

5. Experiments and Analysis

5.1. Experimental Results

The experimental environment consists of a single server, which has an E3-1220 CPU with 3.1Hz, 2GB of main memory, and is running on the Windows Server2003 operating system.

We compare stream model and traditional relation model. Table1 provides a summary of the time elapse when we query for results using 50 runs, including the time elapsed before the first query result is returned, the time elapsed before the first query result is returned, and the time total time.

Table 1. Computation launch

Operation	Mean time of Stream Model(ms)	Mean Time of Relation Model(ms)
First query result	357.014	247.739
Last query result	1522.157	2417.682
Computation time	1605.028	2518.349

5.2. Experimental Results

Figure 4 illustrates the CPU load of the different reliability strategies. The experiments show that there is measurable CPU overhead due to Stream Model and the Relational Model, while Figure 5 illustrates the failure rate of the stream Model and the Relational Model. The experiments show that the both failure rate and CPU Load of the Stream Model is preferable to that of the Relational Model.

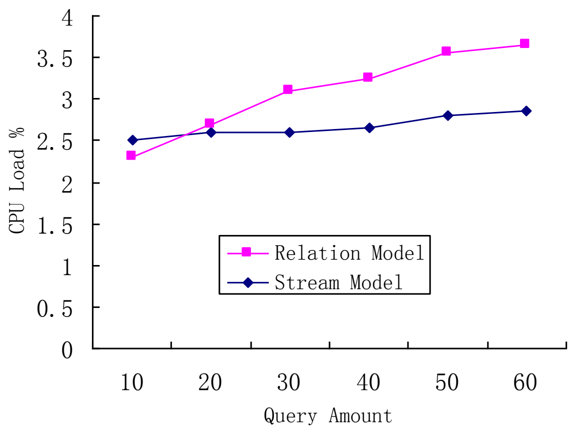

Figure 4. The Comparison of CPU load Rate

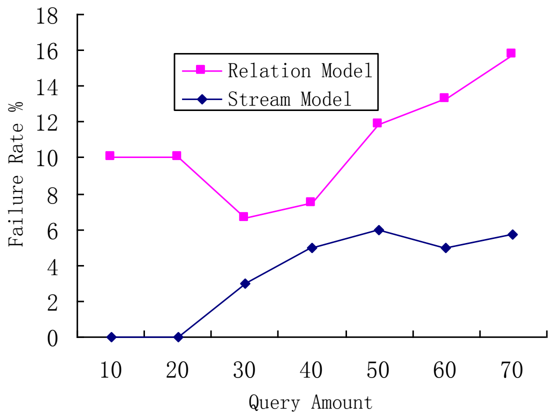

Figure 5. The Comparison of Failure Rate

5. Conclusion

Sensor systems and Internet of Things offer new opportunities for building applications that have information and control in HRIS previously very difficult or even inaccessible. After the acquisition of the human information, a major problem of data storage and exploitation arises, particularly for systems that dealing with real-time. The manuscript has contributed in the construction of the specific building which is not easy because the data management techniques used in traditional databases are not generally suitable for sensor networks because of their specificities.

In general, HR management had undergone radical changes over the last fifty years, while technological development has enabled the transformation of many business activities. The importance of HRIS system is multifaceted, ranging from operational assistance in collecting, storing and preparing data for reports, simplifying and accelerating the processes and controlling the available data. In the future, we can develop more other application systems in HRMS field based on DSMSs for improving the performance and really reaching real-time.

References

[1] Beadles, C. Lowery, and K. Johns, "The Impact of Human Resource Information Systems: An Explarotary Study in the Public Sector", Communications of the IMMA, vol. 5, no. 5, **(2005)**.
[2] Dainty, Loosemore and Lingard, "Human Resource Management in Construction Projects Strategic and Operational Approaches", Taylor and Francis **(2003).**
[3] B. Babcock, S. Babu, M. Datar, R. Motwani and J. Widom, "Models and issues in data streams. In: Popa L, ed. Proc. of the 21st ACM", **(2002).**
[4] R. Avnur and H. J. Eddies, "Continuously adaptive query processing", In: Chen W, Naughton JF, Bernstein PA, eds. Proc. of the 2000 ACM SIGMOD Int'l Conf. on Management of Data, Dallas: ACM Press, **(2000)**, pp. 261-272.
[5] D. Carney, U. Cetinternel, M. Cherniack, C. Convey, S. Lee, G. Seidman, M. Stonebraker, N. Tatbul and S. Zdonik, "Monitoring streams: A new class of DBMS applications", Technical Report, CS-02-01, Providence: Department of Computer Science, Brown University, **(2002)**.
[6] S. Guha, N. Mishra, R. Motwani and L. O'Callaghan, "Clustering data streams", In: Blum A, ed. The 41st Annual Symp. on Foundations of Computer Science, FOCS 2000. Redondo Beach: IEEE Computer Society, **(2000)**, pp 359-366.
[7] P. Domingos and G. Hulten, "Mining high-speed data streams", In: Ramakrishnan R, Stolfo S, Pregibon D, eds. Proc. of the 6th ACM SIGKDD Int'l Conf. on Knowledge Discovery and Data Mining, Boston: ACM Press, **(2000)**, pp.71-80.
[8] T. Johnson, "A Heartbeat Mechanism and Its Application in Giga-scope," Proc. 31st Conf. Very Large Databases (VLDB 05), VLDB Endowment, **(2005),** pp. 1079-1088.
[9] P. A. Tucker, "Exploiting Punctuation Semantics in Continuous Data Streams," IEEE Trans. Knowledge and Data Eng., vol. 15, no. 3, **(2003)**, pp. 555–568.
[10] L. Ding and E. A. Rundensteiner, "Evaluating Window Joins Over Punctuated Streams," Proc. 13th

ACM Int'l Conf. Information and Knowledge Management, (CIKM 04), ACM Press, **(2004)**, pp. 98-107.

[11] U. Srivastava and J. Widom, "Flexible Time Management in Data Stream Systems," Proc. 23rd ACM SIGMOD-SIGACT-SIGART Symp. Principles of Data-base Systems, (PODS 04), ACM Press, **(2004)**, pp. 263-274.

[12] H. Balakrishnan, "Retrospective on Aurora," Very Large Databases J., vol. 13, no. 4 **(2004)**, pp. 370-383.

[13] Y. Bai, "Optimizing Time Stamp Management in Data Stream Man-agement Systems," Proc. 23rd IEEE Int'l Conf. Data Engineering., (ICDE 07), IEEE CS Press, **(2007)**, pp. 1334-1338.

[14] S. Chakravarthy and D. Mishra, "Snoop: An Expressive Event Specifcation Language for Active Databases," Data Knowledge Engineering., vol. 14, no. 1, **(1994)**, pp. 1-26.

[15] Y. Bai, "A Flexible Query Graph-Based Model for the Effcient Execu-tion of Continuous Queries," Proc. 23rd IEEE Int'l Conf. Data Eng., (ICDE 07), IEEE CS Press, **(2007)**, pp. 634-643.

[16] Y. Bai, "RFID Data Processing with a Data Stream Query Language," Proc.23rd IEEE Int'l Conf. Data Eng., (ICDE 07), IEEE CS Press, **(2007)**, pp. 1184-1193.

[17] M. G. Martinsons, **"**Benchmarking human resource information systems in Canada and Hong Kong", [doi: 10.1016/0378-7206(94)90028-0], Information & Management, vol. 26, no. 6, **(1994),** pp. 305-316.

[18] V. Y. Haines, and A. Petit, "Conditions for successful human resource information systems", Human Resource Management, vol. 36, no. 2. **(1997)**, pp. 261-275.

[19] Z. Hussain, J. Wallace, and N. E. Cornelius, "The use and impact of human resource information systems on human resource management professionals", Information and Management, vol. 44, no.1. **(2007)**, pp. 74-89

[20] E. Ngai, and F. Wat, "Human resource information systems: a review and empirical analysis", Personnel Review, vol. 35. no. 3, **(2006)**, pp. 297-314.

[21] M. Z. Razali, and D. Vrontis, "The Reactions of Employees toward the Implementation of Human Resources Information Systems (HRIS) as a Planned Change Program: A Case Study in Malaysia", Journal of Transnational Management, vol. 15. no 3, **(2010)**, pp. 229-245.

[22] C. Tansley, and S. Newell, "A knowledge-based view of agenda-formation in the development of human resource information systems", Management learning, vol. 38. no 1, **(2007)**, pp. 95-119.

[23] V. Browning, F. Edgar, B. Gray, and T. Garrett, "Realising competitive advantage through HRM in New Zealand service industries", The Service Industries Journal, vol. 29, no. 6, **(2009)**, pp. 741-760.

[24] T. S. H. Teo, G. S. Lim, and S. A. Fedric, "The adoption and diffusion of human resources information systems in Singapore", Asia Pacific Journal of Human Resources, vol. 45, no. 1, **(2007)**, pp. 44-62.

[25] I. Troshani, C. Jerram, and S. R. Hill, "Exploring the public sector adoption of HRIS", Industrial Management and Data Systems, vol. 111, no. 3, **(2011)**, pp. 470-488.

[26] V. Vo, L. Jiawei and B. Vo, "Stream Time Series Approach for Supporting Business Intelligence [J]", International Journal of Database Theory and Application, vol. 6, no. 2, **(2013),** April, pp. 1-18.

[27] N. Venkateswaran, "e-HRM, Department of Management Studies Panimalar Engineering College Chennai", available on: http://www.indianmba.com/Faculty_Column/FC555/fc555.html, (**2007**).

[28] H. Balakrishnan, M. Balazinska, D. Carney, U. Cetintemel, M. Cherniack, C. Convey, E. Galvez, J. Salz, M. Stonebraker, N. Tatbul, R. Tibbetts and S. Zdonik, "Retrospective on Aurora", VLDB Journal vol. 13, no. 4, **(2004)**, pp. 370-383. doi:10.1007/s00778-004-0133-5.

Author

Wei Chen, she received her bachelor 's degrees from Chongqing University of Posts and Telecommunications in 2003, and her master's degree from Chongqing University of Posts and Telecommunications in 2007. She works as lecturer of Chongqing University of Posts and Telecommunications. Her main research area includes Human Resource Information Management.

