УДК [004.89:004.422.61]:004.2

Вергунов В.В.[1], студент магістр

Інтелектуальні системи збереження даних як інструмент для підтримки оптимальної структури даних та архітектури бази даних

У статті розглядається інтелектуальна система збереження даних як об'єднання реляційної моделі та NoSQL моделі для оптимізації структур баз даних, їхніх представлень та запитів до баз даних.

Наведено приклад архітектури такої системи для комп'ютерних систем з декількома джерелами даних, які мають різну швидкість взаємодії з даними та можливістю масштабування їх кількості.

Ключові слова: реляційна модель, NoSQL модель, інтелектуальні системи.

[1]Київський національний університет імені Тараса Шевченка, 03680, м. Київ, пр-т Глушкова 4д, e-mail: victor.verg@hotmail.com

Vergunov V.V.[1], master student

Intelligent data storage systems as tool for support in an optimized data structure and database architecture

This article is represented an intelligent storage system as association relational model and NoSQL models for optimizing database structures, their representations and queries to databases.

In this article represented example of such system architecture for computer systems with multiple sources of data, which have different speed of interaction with data and ability to scale the number of sources.

Key Words: relational model, NoSQL model, intelligent systems.

[1]Taras Shevchenko National University of Kyiv, 03680, Kyiv, Glushkova st., 4d,
e-mail: victor.verg@hotmail.com

Статтю представив д.ф.-м.н., проф.Буй Д.Б.

At this moment of time is possible choose only one of the database models with whole features one of them but losing whole features of another model [1]. There are some attempts in this area, such like OLAP (online analytical processing) [2]. For firstexplore the two models and determine which properties are key features. Then compare those features with OLAP model for full understanding what featurs are missing. And implement those features in intelligent data storage system and bind with computer architecture for best performance.

The relational model for database management is a database model based on first-order predicate logic, first formulated and proposed in 1969 by Edgar F. Codd. In the relational model of a database, all data is represented in terms of tuples, grouped into relations. A database organized in terms of the relational model is a relational database.The purpose of the relational model is to provide a declarative method for specifying data and queries: users directly state what information the database contains and what information they want from it, and let the database management system software take care of describing data structures for storing the data and retrieval procedures for answering queries.

NoSQL databases are often compared by various non-functional criteria, such as scalability, performance, and consistency. This aspect of NoSQL is well-studied both in practice and theory because specific non-functional properties are often the main justification for NoSQL usage and fundamental results on distributed systems like the CAP theorem apply well to NoSQL systems. At the same time, NoSQL data modeling is not so well studied and lacks the systematic theory found in relational databases.

OLAP overview

OLAP tools enable users to analyze multidimensional data interactively from multiple perspectives. OLAP consists of three basic analytical operations: consolidation (roll-up), drill-down, and slicing and dicing [3]. Consolidation involves the aggregation of data that can be accumulated and computed in one or more dimensions. For example, all sales offices are rolled up to the sales department or sales division to anticipate sales trends. By contrast, the drill-down is a technique that allows users to navigate through the details. For instance, users can view the sales by individual products that make up a region's sales. Slicing and dicing is a

feature where by users can take out (slicing) a specific set of data of the OLAP cube and view (dicing) the slices from different viewpoints.

There are three main types of OLAP systems [4]:

- Multidimensional (MOLAP) is the 'classic' form of OLAP and is sometimes referred to as just OLAP. MOLAP stores this data in an optimized multi-dimensional array storage, rather than in a relational database. Therefore it requires the pre-computation and storage of information in the cube - the operation known as processing. MOLAP tools generally utilize a pre-calculated data set referred to as a data cube. The data cube contains all the possible answers to a given range of questions. MOLAP tools have a very fast response time and the ability to quickly write back data into the data set.
- Relational (ROLAP) works directly with relational databases. The base data and the dimension tables are stored as relational tables and new tables are created to hold the aggregated information. Depends on a specialized schema design. This methodology relies on manipulating the data stored in the relational database to give the appearance of traditional OLAP's slicing and dicing functionality. In essence, each action of slicing and dicing is equivalent to adding a "WHERE" clause in the SQL statement. ROLAP tools do not use pre-calculated data cubes but instead pose the query to the standard relational database and its tables in order to bring back the data required to answer the question. ROLAP tools feature the ability to ask any question because the methodology does not limit to the contents of a cube. ROLAP also has the ability to drill down to the lowest level of detail in the database.
- Hybrid (HOLAP). There is no clear agreement across the industry as to what constitutes "Hybrid OLAP", except that a database will divide data between relational and specialized storage. For example, for some vendors, a HOLAP database will use relational tables to hold the larger quantities of detailed data, and use specialized storage for at least some aspects of the smaller quantities of more-aggregate or less-detailed data. HOLAP addresses the shortcomings of MOLAP and ROLAP by combining the capabilities of both approaches. HOLAP tools can utilize both pre-calculated cubes and relational data sources.
- Real-Time (RTOLAP) [5]. Whilst many OLAP Servers like Microsoft Analysis Services store pre-calculating consolidations and calculated elements to achieve rapid response times. A Real Time OLAP Server will calculate the values on the fly, when they are required. The essential characteristic of RTOLAP system is in holding all the data in RAM.It is a protocol which analyzes fly values when required. It saves every bit of information in RAM. The calculations are executed in a “right-away” manner which reduces the setback linked with “information outburst” since it only saves information under the RAM size standard.

NoSQL conceptual models

First, should be note that SQL and relational model in general were designed long time ago to interact with the end user. This user-oriented nature had vast implications:

- The end user is often interested in aggregated reporting information, not in separate data items, and SQL pays a lot of attention to this aspect.
- No one can expect human users to explicitly control concurrency, integrity, consistency, or data type validity. That’s why SQL pays a lot of attention to transactional guaranties, schemas, and referential integrity.

On the other hand, it turned out that software applications are not so often interested in in-database aggregation and able to control, at least in many cases, integrity and validity themselves. Besides this, elimination of these features had an extremely important influence on the performance and scalability of the stores. And this was where a new evolution of data models began:

- Key-Value storage is a very simplistic, but very powerful model. Many techniques that are described below are perfectly applicable to this model.
- One of the most significant shortcomings of the Key-Value model is a poor applicability to cases that require processing of key ranges. Ordered Key-Value model overcomes this limitation and significantly improves aggregation capabilities.
- O rdered Key-Value model is very powerful, but it does not provide any framework for

value modeling. In general, value modeling can be done by an application, but BigTable-style databases go further and model values as a map-of-maps-of-maps, namely, column families, columns, and time stamped versions.

- Document databases advance the BigTable model offering two significant improvements. The first one is values with schemes of arbitrary complexity, not just a map-of-maps. The second one is database-managed indexes, at least in some implementations. Full Text Search Engines can be considered a related species in the sense that they also offer flexible schema and automatic indexes. The main difference is that Document database group indexes by field names, as opposed to Search Engines that group indexes by field values. It is also worth noting that some Key-Value stores like Oracle Coherence gradually move towards Document databases via addition of indexes and in-database entry processors.
- Finally, Graph data models can be considered as a side branch of evolution that origins from the Ordered Key-Value models. Graph databases allow one model business entities very transparently (this depends on that), but hierarchical modeling techniques make other data models very competitive in this area too. Graph databases are related to Document databases because many implementations allow one model a value as a map or document.

General Notes on NoSQL:

- NoSQL data modeling often starts from the application-specific queries as opposed to relational modeling. Relational modeling is typically driven by the structure of available data. NoSQL data modeling is typically driven by application-specific access patterns, i.e. the types of queries to be supported.
- NoSQL data modeling often requires a deeper understanding of data structures and algorithms than relational database modeling does.
- Data duplication and denormalization are common.
- Relational databases are not very convenient for hierarchical or graph-like data modeling and processing. Graph databases are obviously a perfect solution for this area, but actually most of NoSQL solutions are surprisingly strong for such problems.

Main requirements for modern databases

At present appeared the so-called phenomenon “Big Data”. Generally accepted opinion that data is now so much that process them with traditional database and program methods are difficult or impossible. This caused a wave of non-relational databases (NoSQL), in which the emphasis is on high scalability.They have the three most important features that are not present at the relational databases:

- Fast access to data.
- Fast replication (data distribution across multiple nodes).
- Flexible (new columns can be add immediately).

There are a few key things that made relational databases most popular databases in the world:

- Supported modeling of relationships and business logic;
- Query Languages;
- Triggers;
- Working with large amounts of data that does not fit in RAM.

Most features of ROLAP can be seen in the list of relational and NoSQL models. Those that have not listed mainly necessary only for Business Intelligence. It is much more concerned with data analysis rather than with data storage and level of data availability. That's why will be considered that all ROLAP features for data storage and availability of this data are represented in association of relational and NoSQL model.

General assumption about system

For the adequacy of the system will be make several assumptions:

- There areminimum three storages with different levels of capacity and availability.
- System have full control on the file system of the storage devices.
- Different data have different numbers of queries to build sample based on their (requests for data are different with different intensity).
- Data are inhomogeneous.
- There are a large number of users using the same data in different representations.

- Assume that with more than 90 percent probability is possible to predict the demand for data [7], [8].

Based on of the foregoing is possible to begin description of the system. Just wanted to point constant on research and attempts to build the hybrid system as the direction of the development of systems forstorage and management data. Storage and management of data becomes every day is much more important and challenging than ever before.

Also, like to point out that the assumption as stated above about the fact that forecasts deemed sufficient as possible [9], because there are quite a few approaches for qualitative solution of the problem [10]. Will only use the results of the existence of such algorithms. In addition, it can be seen as a system of mass service with three type's devices, with infinitely requests and known limitations on the queue for each type of device. This is common task for theory of querying that why it would not be considered. It will only used as an element of the system acceleration and allow it to function as OLAP.

General descriptionof system architecture

In Figure 1 can be seen the general scheme of architecture and its interaction with users.

The figure shows that for storing data uses three types of devices:

- Main storage, it acts as the main storage for the data. It can include any number of nodes. I.e. it is considered as one but it can be the network of devices.

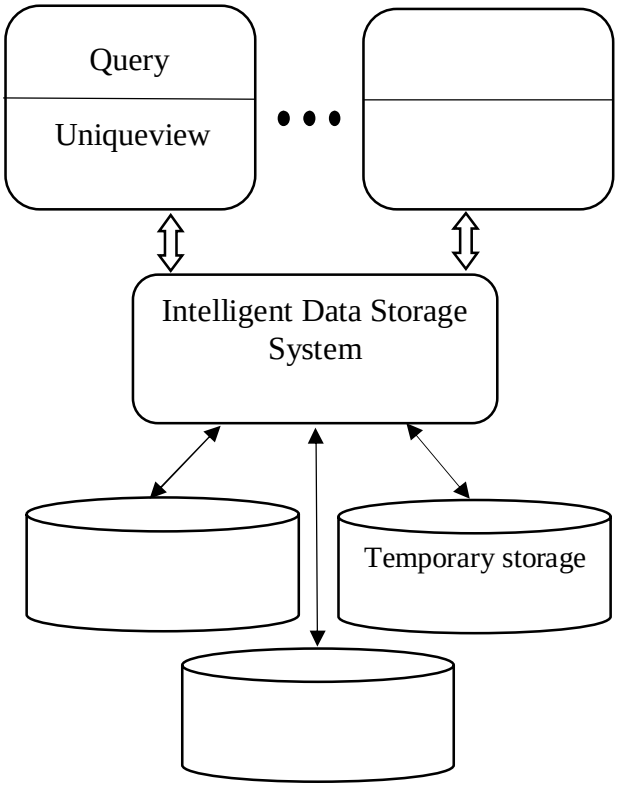

Fig. 1 Systemsgeneralscheme.

- Temporary storage is storage that can also consist of multiple physical devices. They have a much better availability than primary storage. Also, have a smaller capacity. I.e. acts as a buffer store. It can be a RAID or SSD, but cannot be RAM (reasons for this will be explained below).
- Control storage are storing transactions and archives stats. It must maintain the integrity to ensure control and monitor data redundancy. As well as possible supports work in atomic operations.

Also on the scheme is Intelligent data storage System. This is module that controls the whole process of data access, organization their storage and constructing representations [9]. Thus is considered that it included ability to predict queries [10]. Completely it will be described below.

In addition, on the scheme there is no conceptual level. It just would not be. To get the maximum flexibility in the management of data, user will have only that views what he needs [11], [12]. This means that not exist hard schema for the data itself. It also means that data cannot follow the rules of normal forms. But still will not complete distancing from these rules. Simply complete adherence to these rules makes it impossible to use approaches of NoSQL model. Nevertheless, completely abandon them are not possible because it means loss ability to use Data Definition Languages.

Construction of Main storage

The main store is storage that stores all the data. I.e. it is a target for the all data in the system. In some cases, not all data may be located in Main storage (small part may be in the Fast storage). The reasons of this situation will been explained later but all of the data still should get to the Main storage, it is the end point for all data in the system.

The basis for the physical representation of data will be tables and linked lists. Since there is full control over the file, system possible to use absolute addresses. That is in the linked list, we can link directly to the physical address of the next element in linked list. Get something similar on a linked list with gaps. The complexity of finding it a little bit better than the average of a linked list. But still worse than in table by unique index.In order to find out which of the types of data will be slower in the search, it is necessary carry out tests of both data structures. Only then possible say for sure.

Find the physical address is quite simple. There is also an opportunity to streamline transactions for optimal use HDD. How it will effected on the transactions will described below.Because even at normal difficulty of the search and reading on it imposing that if the cell is inside cylinders, speed of reading it falling. According to this table with bigger elements will be on the outside of the cylinders.CHS (cylinder/head/sector) tuples can be mapped to LBA address with the following formula:

$$LBA = ((C \times HPC) + H) \times SPT + S - 1$$

Where:

- , H and S are the cylinder number, the head number, and the sector number
- BA is the logical block address
- PC is the maximum number of heads per cylinder (reported by disk drive, typically 16 for 28-bit LBA)
- PT is the maximum number of sectors per track (reported by disk drive, typically 63 for 28-bit LBA)

LBA addresses can be mapped to CHS tuples with the following formulas:

$$C = LBA \div (SPT \times HPC)$$

$$H = (LBA \div SPT) \bmod HPC$$

$$S = (LBA \bmod SPT) + 1$$

Construction of Temporary storage

This storage is building on the principle of key-value database, something like Berkley DB.Quite a lot of databases is building on them. Search key with value already are implementing in different ways, through the trees or hash tables, etc.

It exists to ensure the atomicity of operations and accelerate the system. Atomicity creates by creating elements in the Temporary storage with entering the mark in the Control storage. After the elements have been put in the Main storage. Will be check in the integrity of the elements. Only after this operation, elements will be removed from the Temporary storage and then removing mark from Control storage. As mentioned possible to predict with high accuracy requirements. So we can put there views which will soon be called.

Construction of Control storage

This storage serves storage views (as a scheme to build them based on the data) and the archives states.

Archives can have four states:

- ormalizated archive, that why executing all Codd rules. Possible to say that this is simple relational table state [14].
- enormalizated archive, this is the archive where do no executing at least one Codd rules and archive is not in any of the four normal forms state [15].
- ot prepared archive it is archive, which in this moment of time does not available for querying. It might changing scheme but in that, moment possible to made query to unchanged archive (before changing of scheme). In addition, it can be replicate of some archive.
- ggregated. It is archive, which already prepared to by view for some user or group of users. Such archive always have archive perent from which it created (Normalized or Denormalizated archive).

A very important task, which relied on Control storage, is monitoring relationships for protection from the consequences of breaking the Codd's rules.

Main tasks designated for the Control storage:

- racking data redundancy;
- ropaganding updates;
- ransaction;
- ata integrity;
- ata independence.

All this things are done by the marks of states. I.e. if there is a failure of Codd rules, they are marked and monitored.

Organization concept of transaction

The nature of the transaction will not explain. Just proceed to the description of how it works in the proposed system.The main purpose of transactions guarantee atomic operation. In cases of this are added the additional verification if the archive is not in the normalization condition. In cases of denormalizated, data transaction occurs as references replacement after checking the data equivalence.

Just knowing the physical address on the cylinder possible order elements of the transaction in such

way, which will accelerate its execution. Maximum overcome Seek Time. For HDD with averagerotation speed 7200 Seek Time is 4,2 ms. At such approach Seek Time will always minimal.

Description ofIntelligent data storage systems like management node

For optimal control of all the processes necessary Intellectual system that will predict, prepare and forward the packet to the nodes [8]. Of course should be possible for user set the whether his view require aggregation or every time a new sample. Besides, the package and query management system should work on improving transactions. Sort elements in it if it will increase speed up.This will require the preservation of different types of query statistics. However, how will be selected the weightswhich defining relevance is question of specific realization or system configuration.It is important that managing task is solved. Moreover, have a good description of solutions this task in many publications. The prediction results are using not only to control the data issue and storage, but also for managing their view. Because, for example by varying the amount of data can be completely rebuilt database schema.

Conclusion

System prototypewas created but without including the intellectual part. It includes only the Main storage and Control storage. The system worked on emulate operating system. HDD for accommodating a Main storage also emulated but like separated area, that why it have own physical addressing field of virtual drive. For the physical storage of data used key value library (Berkeley db can redistribute as library). Further established two data structures as described above. For Control storage used RAM.

Tests have shown that in the case of tables, the search speed is slightly better than in mixing data structure, but very close. However, on join tables best result was shown by mixed data structure. In addition, it is possible make the assumption that Seek Time equally influenced on the sample construction speed in both cases.

List of used sources:

1. *Date, C. J.* (2007). An Introduction to Database Systems (8 ed.). Boston [u.a.]: Pearson Education. ISBN 0-321-19784-4.
2. *Codd E.F., Codd S.B., and Salley C.T.* (1993). "Providing OLAP (On-line Analytical Processing) to User-Analysts: An IT Mandate". Codd& Date, Inc. Retrieved 2008-03-05.
3. *O'Brien&Marakas.* OLAP2011, p. 402-403.
4. *Nigel Pendse* (2006-06-27). OLAP architectures. Retrieved 2008-03-17.
5. *Bach Pedersen, Torben; S. Jensen, Christian* (December 2001). "Multidimensional Database Technology". Distributed Systems Online (IEEE): 40–46. ISSN 0018-9162.
6. *Owen Kaserand Daniel Lemire,* Attribute Value Reordering for Efficient Hybrid OLAP, Information Sciences, Vol. 176, Issue 16, p. 2279-2438, 2006.
7. *Dinda P.* (2001) On line Prediction of the Running Time of Tasks. In: Proceedingsof 10-th IEEE International Symposium on High Performance Distributed Computing, p. 383-394.
8. *Zijiang Yan,* Researchon Server Load Prediction Based on Wavelet Packet Theory, 23-25 Nov. 2007, p.610 – 613.
9. *Khotanzad A., Sadek N.* Multi-scale high-speed network traffic prediction using combination of neural networks. Proceedings of the International Joint Conference on Neural Networks, 2003, p. 1071-1075.
10. *Skicewicz J., Dinda P., Schopf J.M.* Multi-resolution resource behavior queries using wavelets [A]. High Performance Distributed Computing, 2001 [C]. Proceedings: 10th IEEE International Symposium on 7-9 Aug. 2001, p. 395-405.
11. *Rachel Pottinger, Philip A. Bernstein* Schema merging and mapping creation for relational sources. EDBT '08 Proceedings of the 11th international conference on Extending database technology: Advances in database technology, p. 73-84, ISBN: 978-1-59593-926-5.
12. *Peter Buneman, Susan B. Davidson, Anthony Kosky.* Theoretical Aspects of Schema Merging. EDBT '92 Proceedings of the 3rd International Conference on Extending Database Technology: Advances in Database Technology, p. 152-167, ISBN:3-540-55270-7.
13. *Codd, E.F.*Further Normalization of the Data Base Relational Model. Presented at Courant Computer Science Symposia, Series 6, "Data Base Systems", N. York City, May 24–25, 1971. IBM Research Report RJ909 (Aug. 31, 1971).
14. *C.J. Date, Hugh Darwen, Nikos Lorentzos.* Temporal Data and the Relational Model. Morgan Kaufmann, 2002, p. 176-181.

Надійшла до редколегії 26.02.2013