The approach to "Big Data" keeping with effective access in multi-tier storage

Globa L.S., Filimonov N.
Information and Telecommunication Department
Institute of Telecommunication Systems
National Technical University of Ukraine «Igor Sikorsky Kyiv Polytechnic Institute»
Kyiv, Ukraine
lgloba@its.kpi.ua

***Abstract*—Big Data has too much cumulative amount of weakly linked data availability of a large amount of structured and unstructured data, their substantial diversity. Such data is handled by means of horizontally scalable software tools and alternative traditional database management systems as well as Business Intelligence class systems. One of the challenges of this Big Data approach is the need to effectively control and process data flows.**

The paper discusses the approach of distributed data storage in multi-tier repositories, taking into account the interdependence of IT infrastructure and location of business objects, as well as the importance and reliability of data over time. The task of placing blocks of data in multi-tier repositories is formulated so that the average access time to them is minimal. A method for solving this problem using a genetic algorithm, the effectiveness of which is confirmed by the results of mathematical modeling is proposed.

***Keywords*—distributed Big Data storing, data storage system, database management system, datacenter.**

I. Introduction

Every year, as a result of human and business activities, more and more data are generated that need to be stored, processed, and analyzed. The vast accumulation of loosely coupled data has been called Big Data, which means that there is a significant amount of structured and unstructured data and a significant variety of them. This data can be processed by means of horizontally scalable software tools used recently and by alternative traditional database management systems, as well as Business Intelligence class systems [1] [2].

Big Data characteristics define at least "three V": volume – physical volume of data, velocity – both speed of increasing volume of data, and necessity of high-speed processing with obtaining results, variety – possibility of simultaneous processing of various types of structured and semi-structured data.

The emergence of large volumes of data storages, Cloud and Frog technologies, the development of the concept of central data storage in certain subject areas of knowledge and business with the simultaneous possibility of virtualization of such a "logical" centralized storage, while physically it is separated by unknown geographically distributed computing datacenters, has exacerbated data management issues. There are problems with reliability – veracity ("fourth V" proposed by IBM), viability – ("fifth V"), value in terms of business – value ("sixth V"), as well as variability – the ability of data to change constantly and visualization – the ability of quick perception the meaning of the results obtained by the user ("seventh and eighth V").

To overcome Big Data challenges, tools of mass-parallel processing with indefinite structure by means of NoSQL database management systems, MapReduce algorithms, and Hadoop [3] are used. Various analytical software components and technologies, approaches to structuring, clustering, and data grouping, that would provide more or less similar characteristics of processing of large data sets, began to refer to Big Data technologies.

It is obvious that the process of constant accumulation of data complicates the work with them. That is why the task of accessing such logically centralized, and physically geographically dispersed, data storages with an uncontrolled end-user data architecture is a daunting problem related to the control and management of information flows by the end-user of data precisely in terms of the requirements of certain processing procedures [4]. The implementation of this approach to Big Data processing covers the following major areas of data processing improvement: databases (both storage structures and processing approaches), mathematical foundations for analytics, the visualization of significant volumes in clear and fuzzy concepts, architecture and flow processing.

The classic approach to storing and processing data no longer fulfills its functions. One of the most urgent tasks is the development of new and modification of old methods that adapt the DSSs (data storage systems) to more stringent conditions [5,6].

The data storage system is a complex software and hardware solution for the organization of reliable storage of information resources and providing guaranteed access to them [7].

In the process of working with data in the DSS the following common properties can be determined:

- accessibility
- capacity
- scalability
- security
- manageability

However, the individual stored information has its own static and dynamic properties. These include value, intensity of usage, and others depending on the tasks. Values and dynamics of change of values define a life cycle of the data manage of which determines efficiency of use of the stored information [8].

The paper is structured as follows: Section 2 contains a state-of-the-art approach to distributed Big Data storing in tiered data storages. Section 3 introduces the proposed approach to distributed Big Data storing in tiered data storages in the requirements of processing procedures terms and places. Section 4 presents the results of the proposed approach to designing distributed Big Data storing in tiered data storages based on data access intensity. Section 5 includes a summary and outlook on future work.

II. State of the art and background

In order to securely store information resources and provide them with guaranteed access when using remote, decentralized, modern data storage systems, lifecycle management processes called ILM (Information Lifecycle Management process) are used, and systems that support data lifecycle management are ILM systems.

The ILM process is a process of data transformation, from the moment you create the data to the deletion, which includes all stages of processing throughout the entire period of work with specific data. ILM systems allow you to organize processes and create data management technologies that allow you to efficiently process and apply specific data throughout their lifecycle. Such systems consider the interdependence of IT infrastructure and the location of business objects, as well as the importance and reliability of data over time. Data that is important to business today may be almost unnecessary tomorrow, called the data aging process. In the case of aging data, the latter can be automatically moved to cheaper and less time-efficient access to the data storage and to use low-performance technologies, depending on their relevance, to save money and improve the use of available data storage resources [9,10].

The implementation of the ILM process includes the following set of actions. First of all, it is the classification of the data itself and the programs for processing them, then – defining a strategy for creating and deleting data, depending on their class, volumes and other parameters of influence on the importance of certain datasets.

A separate aspect of the ILM process is the management of the processes of storing input streams, uploading them to data storages and extracting them. As a logical conclusion, the concept of ILM describes the multi-tier storage of data on different types of media, which, in turn, divided by the parameters of price, speed of access, reliability, etc. [11].

To integrate the ILM process into application systems that support a particular business environment, perform the following actions:

- establish a data storage system
- classify data (tiers, cycles, policies)
- moving between tiers manually
- automate basic business processes
- fully integrate the individual steps of the ILM process into all software components of the application system.

Moving data manually requires the complex work of analysts on each individual implementation of multi-tier DSS, so there is a transition to a DSS architecture with fully automated FAST (Fully Automated Storage Tiering) storage.

For automated multi-tier storage architectures, the DSS hierarchy has the following tiers:

- Tier 1 – designed for business-critical data (mission-critical, business-critical), characterized by minimal access time and high availability. Currently typically implemented on SSD storage systems.
- Tier 2 – online access reference data, sometimes archive systems, document storage systems for which speed of data access and system availability are important, but not critical for business, traditionally the main storage array of the organization. It is implemented on the whole variety of arrays and storage systems, based on hard drives (FC, SAS, SATA, SCSI)
- Tier 3 – mainly archive systems and backup systems. In addition to disk arrays can be implemented including on tape libraries.

The moving between tiers approach is called Tiering. Each of the tiers is characterized by certain properties (price, speed, capacity, security and others). Typically, the tiering process is called a mechanism for moving data between disks of different tiers, or between disks and a magnetic stripe, or ROM storage. Often, it is used to organize the ILM data storage in accordance with the status of the data and the QoS level.

Discs used today in DSSs differ in performance, capacity, cost. And at the same price, productivity and capacity are usually inversely proportional (larger capacity – lower productivity (SATA), lower capacity – greater productivity (SAS), even greater productivity – even smaller capacity (SSD). DSSs are generally characterized by an IOPS/cost ratio where IOPS is the number of I / O operations performed by DSSs per second (input / output operations per second).

In general, a small block of data requires most of the total data processing time, so it is usually determined by the speed of the system. In contrast, large blocks of data refer to so-called "cold" data, which have almost no effect on performance. A data processing center (storage center) can consist of many tiers that use different types of media.

It is well known that when creating all business systems, they require cost reduction without loss of quality. Another way to save, and therefore optimize, is to de-energize individual repositories (racks, disks) with unnecessary data at specific times. For example, night-time shutdown of devices that store “cold” data or data that will not be needed exactly during off-hours. This approach takes into account the additional requirements when the analysis and further tuning of the system must occur in such a way that shutdowns for the sake of economy do not interfere with business processes, and the tuning of the process of de-energization occurred without fatal consequences, because the processing datacenter, apart from the carriers themselves, has configuration servers and other infrastructure.

Based on the foregoing, there is a complex set of problems associated with the organization and control of large dynamic data streams, the collection, storage, processing of large amounts of data, operational support of users for adequate decision making.

For large systems, tier partitioning is quite a challenge because most of the application software is located in the processing datacenter, and the distributed application components are located in several processing datacenters at once. Such an application software infrastructure may use more than one DSS, but several, making it difficult to break down the entire set of DSSs at the tier.

Each datacenter has its own DSSs, application servers, and other infrastructure that allows you to quickly access application software components to information that is situated directly within the DSSs of a particular datacenter. But if the system is distributed, then the application software needs to receive information hosted by different datacenters, and the speed of data access is highly dependent on the distance to other datacenters, speed, response and other network settings.

III. Problem definition for Big Data distributed storing in tiered data storages based on data access intensity

With regard to optimizing the placement of data in distributed systems, it is obvious that the faster the application software receives information from the DSS, the faster it processes the data. To improve the performance of the system, it is advisable to place the data on media that can be accessed as quickly as possible, taking into account the features of the location and processing of the specific application software component that uses them. There is a problem of optimal placement of data in distributed DSSs to improve performance of a particular application component. And, unlike a system consisting of one DSS and one application server, there is a situation where it is more advantageous to keep information on less fast media but less time accessing the application server. In such a system, it is also important to consider the intensity of data usage by individual servers in order to properly calculate the optimal location [12].

For application servers located in a single datacenter, it is possible to break up the DSS, which is physically hosted at different datacenters, at the following criteria:

- Speed of access to storage. The repository acts as a unit.
- Media performance. The disk acts as a unit.
- Media access speed. The disk acts as a unit.

By unit we mean an inseparable part of the tier.

For application servers located in different datacenters:

- The sum of the output of the speed of access to the repository and the intensity of access to the application server with which the exchange is being performed. Storage is a unit.
- The sum of the output of the speed of access to the media and the intensity of calls of the application server with which the exchange is being performed. Storage is a unit.

This is the idea underlying the tiering process. If a particular type of data is active, it is automatically transferred to more modern and faster, performance-enhancing media and less active data – to less productive and outdated disk types, resulting in an IOPS / cost ratio.

IV. Mathematical model for the proposed approach

Consider the case where the functions of application software components are accessed by several blocks, and each block can be accessed by several functions, and also, each data block must be on only one data carrier (see Fig. 1).

We introduce the following notation:

F_k – function calls, $k = 1, \dots, l$;

μ_k – the average intensity of function calls F_k;

D_i – data blocks accessed by the function, $i = 1, \dots, n$;

d_i – volume of data block D_i;

λ_{ki} – the intensity with which the function F_k accesses the data block D_i;

S_j – physical data carriers on which data blocks are stored, $j = 1, \dots, m$;

t_j – the time of receiving data from the S_j repository;

t_{ki} – time to access the F_k function;

s_j – volume of storage S_j;

x_{ij} – a Boolean variable that determines whether the data block D_i is located in the S_i repository;

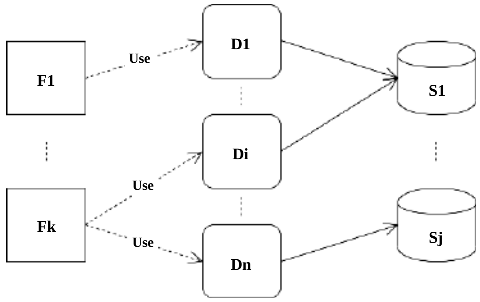

Figure 1. Scheme of organization of function calls of application software to the data of physically distributed DSS

Since each block of data must be on only one data media, the condition must be:

$$\sum_{j=1}^{m} x_{ij} = 1 \qquad (1)$$

where $i = 1, \ldots, n$

Formulation of the problem. In a situation where resources in storage systems are limited and there are no strict requirements for the response time of functions, the task arises to optimally place blocks of data on data carriers so that the average system performance reaches the maximum value and the average access time to the data is minimal.

The average intensity with which the function F_k accesses the data block D_i can be represented as follows:

$$\mu_{D_i} = \sum_{k=1}^{l} \mu_k \lambda_{ki} \qquad (2)$$

and the criteria for minimizing the average access time can be written:

$$T_{min} = min \sum_{i=1}^{n} \sum_{j=1}^{m} \mu_{D_i} \cdot x_{ij} \cdot (t_j + t_{ki}) \qquad (3)$$

Then the problem of minimizing the average access time can be formulated: minimizing the average time of access to the data (3), provided the execution of the constraint (1).

V. The problem solution based on genetic algorithm

The above problem is a linear Boolean programming problem and, with a small number of independent variables, can be effectively solved by using exact methods. When a distributed application infrastructure uses more than one physically distributed DSS that changes dynamically over time, is itself territorially distributed, the dimension of the problem is high, then it is advisable to use a genetic algorithm (GA) [13].

Since each block of data must be on only one data carrier, in order to encode the genes according to the algorithm, it is necessary to move from the Boolean matrix n x m of variables x_{ij} to the vector of discrete variables $y_i \in [1, m]$, in which each element contains a data carrier number S_j, $(j = 1, \ldots, m)$, at which the corresponding block of data is located at the data instant.

Example of gene coding:

$$x_{ij} = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \rightarrow y_i = \begin{bmatrix} 4 \\ 2 \\ 3 \\ 1 \\ 3 \\ 4 \\ 2 \\ 1 \\ 2 \end{bmatrix}$$

It is recommended to use a controlled genetic algorithm to solve problems of this class [14].

The block diagram of the algorithm is shown in Fig. 2.

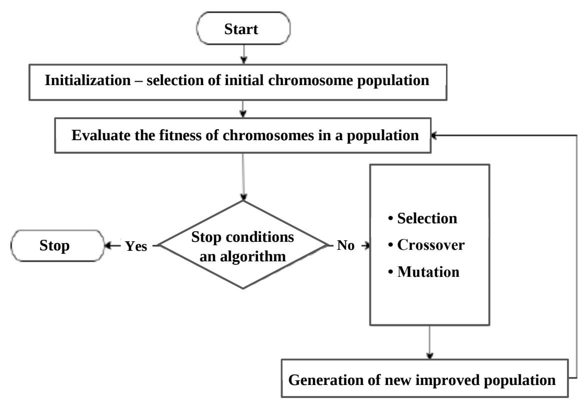

Figure 2. Flowchart of a managed genetic algorithm

Consider the basic steps of executing a managed genetic algorithm:

Step 1: Initialization – selection of the initial chromosome population;

Step 2: Evaluate the fitness of chromosomes in a population;

Step 3: If the end condition is positive then stop, and the choose of the "best" chromosome in current population;

Step 4: Create a new population with repeating next steps:

- Selection – Extract a subset of genes from a population, according to any definition of quality;
- Crossover – crossing parents genes;
- Mutation – forms a new population of offspring;

Step 5: Go to step 2.

VI. The results of proposed solution

The effectiveness of the proposed approach to storing “big data” in multi-tier “logically” centralized repositories in accordance with the tasks of the business environment, which function as physically distributed DSSs, was tested using mathematical modeling tools. Below are graphs that compare the quality of data access (storage speed, media access speed, media performance) and the data volume change graph.

During the experiment, an architecture with automated multi-tier data storage was applied, for which the media was broken down by media types and the average amount of data in the database of a specific fragment of the DSS. The (Fig. 3) depicts the dependence without applying the proposed approach, where the Y axis represents the μk – average intensity of function calls to the DSS, and the X axis shows the average amount of data in the database of a specific node of the DSS.

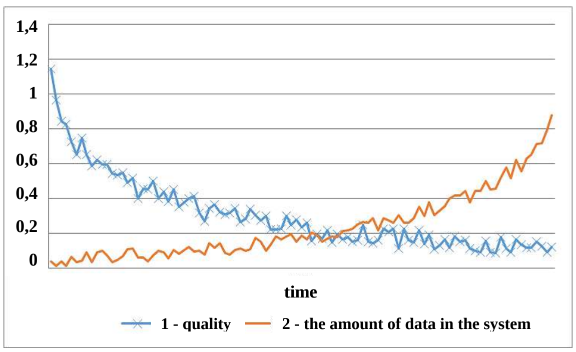

Figure 3. Quality of access to information, depending on the amount of data stored in the system according to their life cycle, without applying the proposed approach

The graph in Figure 3 shows that the quality of access to information in the process of increasing it is not constant and at different points in time changes, leading to low rates of data access speed, which in turn does not meet the requirements of the application systems.

As a result of applying the proposed approach, after structuring and grouping the data, the partitioning of the media at the tier by media types and the average amount of data in the database of a specific node of the DSS, we obtain the following result (Fig. 4).

As a result of dynamic restructuring we get: with the increased amount of information, the quality of access to data at any address remains almost at the same tier, which allows to provide stable access to all information resources while maintaining maximum access efficiency.

VII. Summary and outlook

The paper deals with the pressing issue of efficient placement of data in multi-tier storage systems, which arises when dealing with a large amount of data, limited resources and stringent requirements for storage performance.

The application of dynamic models of optimal data placement, depending on their volume in the system,

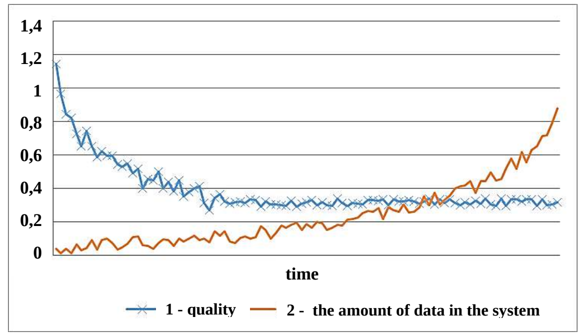

Figure 4. Quality of access to information, depending on the amount of data stored in the system according to their life cycle, using the proposed approach

taking into account their life cycle, allows to increase the performance indicators of the distributed data storage system, which was proved by the results of the experiment. The greatest increase in the efficiency of the functioning of the distributed storage system resulted in the movement between the carriers of different productivity.

It is recommended to apply the proposed model when designing system software for multi-tier data storage systems using partitioning of databases and setting up information lifecycle management (ILM) processes.

Further, there is a need to improve the proposed approach in terms of increasing the diversity of media types, allowing data to be accessed so that access to important and frequently used data is performed as quickly as possible, and outdated data is moved to less rapid media or archived at all.

References

[1] Min Chen, Shiwen Mao, Yin Zhang, Victor C.M. Leung. Big Data. Related Technologies, Challenges, and Future Prospects. – Spinger, 2014. – 100 p. – ISBN 978-3-319-06244-0. – DOI:10.1007/978-3-319-06245-7.

[2] James Manyika et al. Big data: The next frontier for innovation, competition, and productivity (engl.) (PDF). McKinsey Global Institute, June, 2011. McKinsey (9 August 2011).

[3] Preimesberger, Chris Hadoop, Yahoo, 'Big Data' Brighten BI Future (engl.). EWeek (15 August 2011).

[4] "Cisco Global Cloud Index: Forecast and Methodology 2015-2020" Cisco Public.

[5] Mayer-Schönberger V. Big Data: A Revolution That Will Transform How We Live, Work, and Think / V. Mayer-Schönberger, K. Cukier. – Houghton Mifflin Harcourt, 2013. – 240 p.

[6] Marz N. Big Data: Principles and Best Practices of Scalable Realtime Data Systems / N. Marz, J. Warren. – Manning Publications Company, 2015. – 328 p.

[7] Schulz G. Cloud and Virtual Data Storage Networking / G. Schulz. – CRC Press Taylor & Francis Group, 2011. – 370 p.

[8] Horizontal, vertical, and functional data partitioning [Online Resource] // Microsoft – Resource Access Mode: https://azure.microsoft.com/en-us/documentation/articles/best-practices-data-partitioning/ – 2018

[9] Information Lifecycle Management. – 2005. – Resource Access Mode: http://i.bnet.com/whitepapers/sun00.pdf

[10] Colarelli D. Massive arrays of idle disks for storage archives / D. Colarelli, D. Grunwald. // Proceedings of the 2002 ACM / IEEE conference on Supercomputing. – 2002. – P. 1-11.

[11] Information Lifecycle Management for Business Data // Oracle Help Centr. – Resource access mode: http://www.oracle.com/us/026964.pdf. – 2007

[12] Storage systems [Electronic resource]. – Resource Access Mode: http://www.compdoc.ru/peripherals/drives/categorization-storage-systems – 2016

[13] Goldberg D.E. Genetic algorithms in search, optimization, and machine learning / D.E. Goldberg. – Addison-Wesley, 1989. – 412 p.

[14] Telenik SF / S.F. Telenik, OI Roller, M.M. Bukasov, S.A. Androsov // Automation. Automation. Electrotechnical systems and systems. – 2010. – № 1 (25). – P. 106-120.

Подход к хранению «больших данных» с эффективным доступом в многоуровневом хранилище

Глоба Л.С., Филимонов Н.

Большие данные имеют слишком большой совокупный объем слабосвязанных данных, большой объем структурированных и неструктурированных данных, их существенное разнообразие. Такие данные обрабатываются с помощью горизонтально масштабируемых программных средств и альтернативных традиционных систем управления базами данных, а также систем класса Business Intelligence. Однако, обработка больших данных на основе такого подхода требует эффективных как контроля, так и доступа к хранилищам данных, контроля процессов обработки потоков данных.

В статье рассматривается подход распределенного хранения данных в многоуровневых хранилищах, учитывающий взаимозависимость ИТ-инфраструктуры и местоположения бизнес-объектов, а также важность и достоверность данных во времени. Задача размещения блоков данных в многоуровневых хранилищах сформулирована таким образом, чтобы среднее время доступа к ним было минимальным. Предложен метод решения этой задачи с использованием генетического алгоритма, эффективность которого подтверждается результатами математического моделирования.

Received 13.12.2019