Ministry of Education and Science of the Republic of Kazakhstan
Suleyman Demirel University

Tamirlan Suleizhan

Development of a mathematical model of the system for collecting and analyzing data

THESIS

Presented in Partial Fulfillment for the
Degree of Master of Science in Computing Systems and Software
(degree code: 6M060100)
Department of Computer Sciences
Faculty of Engineering and Natural Sciences

Supervisor: **Olimzhon Baimuratov**

Kaskelen, 2019

Abstract

This thesis is aimed at developing practical skills of building mathematical systems for data collection and analysis. The peculiarity of the work is to learn to self-decision-making, to develop research skills, which is especially important in a dynamic world. Data collection is carried out in almost all areas of science and technology. Over the past few years, data collection techniques have been applied to applications and new corporations have emerged to commercialize the technology. Many business and financial problems are successfully solved with the use of big data. Problems of management, classification, pattern recognition, forecasting, inherent in almost all application areas, such as medicine, military Affairs, aviation and space, construction, are increasingly solved with the use of this technology. In this regard, you will see a fundamental understanding of the basic concepts and models of big data, as well as learn how to apply this knowledge in practice.

Аңдатпа

Бұл дипломдық жұмыс деректерді жинау және талдау үшін математикалық жүйелерді құрудың тәжірибелік дағдыларын жасауға бағытталған. Жұмыстың орындалу ерекшелігі-өз бетінше шешім кабылдауды үйрену, зерттеу кабілеттерін дамыту, бұл әсіресе каркынды дамып келе жаткан әлемде маңызды. Деректерді жинау іс жүзінде ғылым мен техниканың барлық салаларында жүзеге асырылады. Соңғы бірнеше жылда деректерді жинау әдістері қолданбалы салаларда колданыла бастады және осы технологияны коммерциялық пайдаланумен айналысатын жаңа корпорациялар пайда болды. Үлкен деректерді пайдалана отырып, бизнес пен қаржының көптеген проблемалары табысты шешілуде. Медицина, әскери іс, авиация және ғарыш, құрылыс спякты барлық колданбалы салаларға тән басқару, жіктеу, бейнелерді тану, болжау міндеттері осы технологияны колдану аркылы жиі шешіледі. Осыған байланысты сіз үлкен деректердің негізгі түсініктері мен модельдері туралы іргелі түсініктерді көресіз, сондай-ақ осы білімді тәжірибеде.

Аннотация

Данная дипломная работа направлена на вырабатывание практических навыков построения математических систем для сбора и анализа данных. Особенностью выполнения работы является то, чтобы научиться к самостоятельному принятию решений, развить в себе исследовательские способности, что особенно важно в динамично развивающемся мире. Сбор данных осуществляется практически во всех областях науки и техники. За последние несколько лет методы сбора данных стали применяться в прикладных областях и появились новые корпорации, занимающиеся коммерческим использованием этой технологии. С использованием больших данных успешно решаются многие проблемы бизнеса и финансов. Задачи управления, классификации, распознавания образов, прогнозирования, присущие практически всем прикладным областям, таким как медицина, военное дело, авиация и космос, строительство, все чаще решаются с применением данной технологии. В связи с этим вы увидите фундаментальное представление об основных понятиях и моделях больших данных, а также научить применять эти знания на практике.

Acknowledgements

I would like to Express my deep gratitude to the reviewer and my supervisor for their comments and very helpful criticism that helped to address the shortcomings in explaining the work at an earlier stage. I also want to thank all the faculty members for their hard work and patience.

Contents

1. Introduction

Big data refers to a wide variety of data sets that cannot be properly processed by traditional applications because of their large size or complex composition. The complexity of the analysis of big data lies in the specifics of their collection, supervision, division, storage, transmission, visualization and confidentiality. Big data analysis is often understood as the application of predictive Analytics or other best practices to extract some useful information from a set of data. Accuracy in big data analysis helps you make better decisions. In turn, making the best decisions can increase production efficiency, reduce costs and reduce risks. Big data Analytics can be applied in areas such as tracking of market conditions, the prevention of the spread of epidemics and the fight against crime. Scientists, CEOs, workers of mass-media and advertising, as well as government agencies often face difficulties in the analysis of huge amounts of data volumes in the areas of search in the Internet, information technology in the field of business and Finance, etc. The work of scientists, especially meteorologists, genome scientists, researchers working in the field of communication, physicists creating complex simulations, and biologists and ecologists is often limited by the processing of huge amounts of data. The amount of data is constantly increasing, as the ability to collect information using low-cost mobile devices, digital aerial photography, cameras, microphones, radio frequency tag readers and wireless sensor networks is expanding[4].

1.1 Analysis and development of big data processing technologies

Let's look at how big data technologies are used in scientific research. A Large Collider was used to capture data from 150 million sensors 40 million times per second. That's about 600 million collisions per second. After filtering and removing more than 99.99995% of them. 100 collisions per second remained. which were studied by researchers. As a result. after studying less than 0.001% of the data from the sensors. the volume of data from all four experiments using the Large Collider was 25 petabytes per year before replication (as of 2012) or nearly 200 petabytes after replication. If all the data from the sensors installed in the Large Collider were recorded. this amount of data would be extremely difficult to process. The data flow exceeded 150 million petabytes per year. or 500 exabytes per day before replication. This is equivalent to 500 quintillion (5 x 1020) bytes per day. which is almost 200 times the amount of data received from all other resources in the world.

1.2 Analysis of big data processing methods

Under the analysis of big data is understood as the analysis of data sets within the capabilities of a personal computer, and within the capabilities of relational database management systems. while both in the first and in the second case in the formation and statistics, and visualization there are certain difficulties. which are the need to ensure coordinated operation of computer programs on tens, hundreds or even thousands of servers. Big data analysis can be characterized by the following parameters:

1. Volume. i.e. the amount of data generated. Whether a particular dataset can be considered big data or not depends on this metric. The data is stored by the SQL server in the cloud.

2. Diversity, i.e. the category to which big data belongs. The knowledge of such membership allows analysts to most effectively work with information.

3. Speed. i.e. the speed at which data are generated or processed in order to

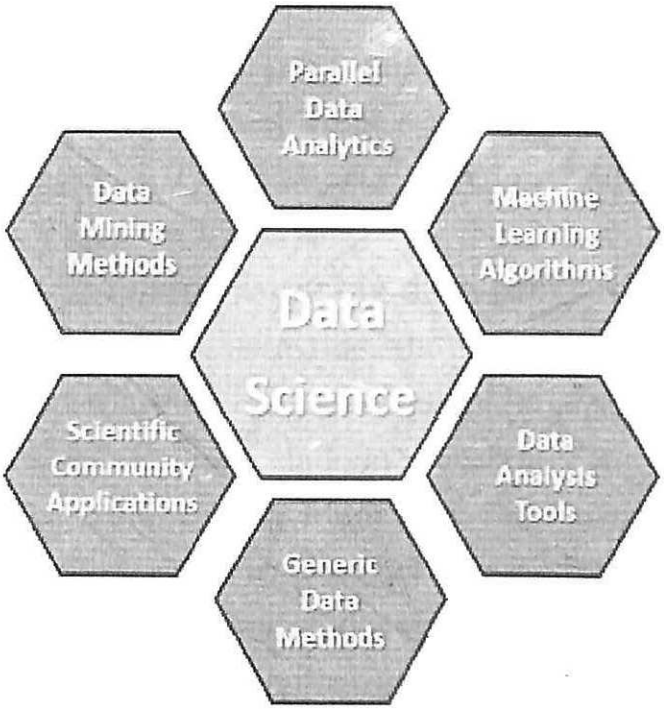


Figure 1.1: Big Data Analysis methods

achieve goals.

4. Variability, i.e., the instability of the data over time.

5. Reliability, i.e. the quality of the data collected, on which the accuracy of the analysis depends.

6. Complexity, i.e. the complexity of the process of correlation and building relationships between data. Let's consider the main methods of big data analysis. Methods of analysis of big data used in modern technologies can be displayed using the following diagram .

Architecture of big data technologies

In 2004, Google published an article on the MapReduce process. The MapReduce framework uses a parallel processing model for very large amounts of data. Within MapReduce, requests are split and distributed across parallel nodes, and they are processed in parallel (allocation stage (IDA)). The results are then collected and delivered (the pre-conversion (Reduce) stage). This framework was very efficient, so other companies also wanted to copy the algorithm. The MapReduce framework was used in an open source Apache project called Hadoop[2]. Recent studies have shown that multi-tier architecture can be successfully used for

big data analysis. The architecture of distributed parallel processing distributes data across multiple nodes. which. producing. respectively. parallel processing. generate the result much faster than other systems. Within this architecture. data is transferred to parallel DBMS using MapReduce and Hadoop frameworks. This involves front-end application servers.

Solving practical problems using big data processing technology Processing of big data now involves, as a rule. the introduction of special software systems, such as hadoop. allowing the processing of large amounts of data on the basis of the concept of Map-Reduce. Hadoop is currently the "de facto" standard for big data processing. Hadoop is a framework that is used to develop applications for big data analysis and visualization. Data storage in this framework is carried out using a special distributed file system HDFS (Hadoop Distributed File System), which is the basis of Hadoop and allows you to store and provide access to data on multiple nodes of the cluster. Thus, if one or more cluster nodes fail. the risk of information loss is minimized and the cluster continues to operate normally.

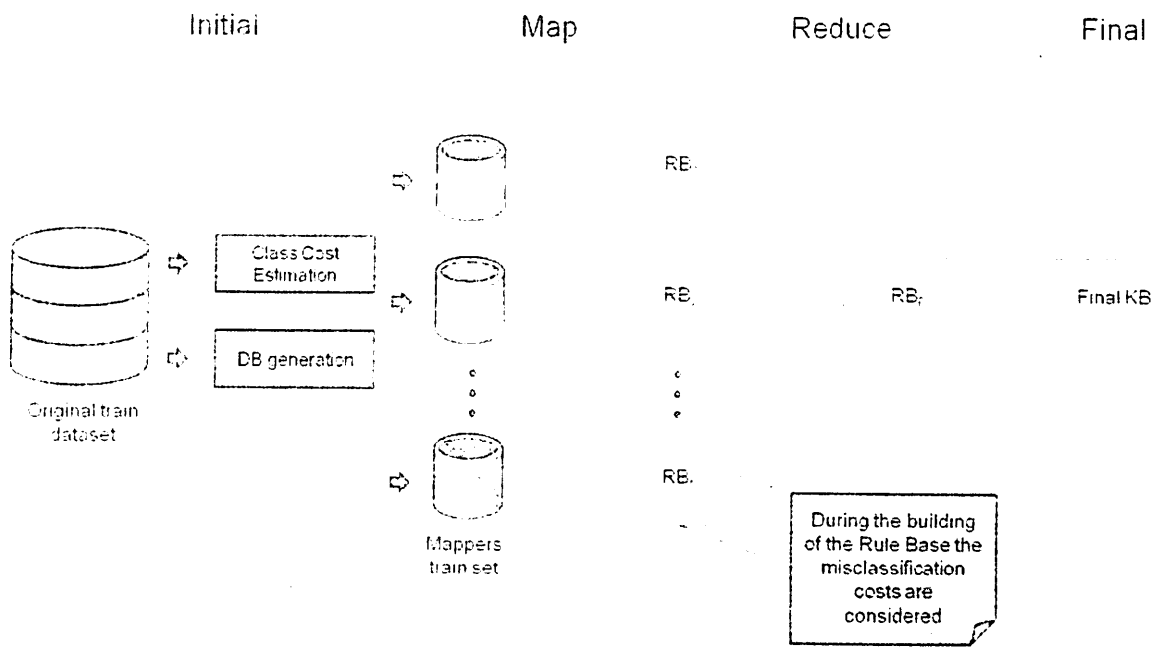


Figure 1.2: Map Reduce Scheme

1.3 Data processing in Hadoop application

Big data uses the MapReduce algorithm, all stages of Map should complete its work before the beginning of Reduce. Also. the input data requires pre-processing. Thus, the General algorithm of work shown in figure 2 is obtained. One of the most urgent tasks of modern information technology is the task of fast processing of large amounts of data. An effective solution to this problem allows you to make faster decisions based on the data obtained in the past. The paper analyzes the methods and approaches to big data processing technology.

2. Data structure

2.1 The concept of data structures and algorithms

Data structures and algorithms are the materials from which programs are built. Moreover, the computer itself consists of data structures and algorithms. The built-in data structures are represented by the memory registers and words where the binary values are stored. The algorithms incorporated in the design of the equipment are rigid rules embodied in electronic logic circuits, according to which the data stored in memory are interpreted as commands to be executed. Therefore, the basis of the work of any computer is the ability to operate with only one type of data - with individual bits, or binary digits. The computer works with these data only in accordance with a constant set of algorithms that are determined by the system of commands of the Central processor.

Tasks that are solved with the help of a computer are rarely expressed in the language of bits. Typically, data is in the form of numbers, letters, texts, symbols, and more complex structures such as sequences, lists, and trees. Even more diverse are the algorithms used to solve various problems: in fact, there are no fewer algorithms than computational problems.

For the precise description of abstract data structures and program algorithms, such formal notation systems are used, called programming languages, in which the meaning of any sentence is determined accurately and unambiguously. Among the tools represented by almost all programming languages, it is possible to refer to the data element using the name assigned to it, or, otherwise, the identifier. Some named values are constants that maintain a constant value in the part of the program where they are defined, others are variables that can be assigned any new value using an operator in the program. But until the program starts running, their value is undefined.

The name of a constant or variable helps the programmer, but it doesn't tell the computer anything. The compiler, which translates the program text into binary code, associates each identifier with a certain memory address. But in order for the compiler to be able to do this, you need to report the "type" of each named value. A person who solves a problem "manually" has an intuitive ability to quickly understand the types of data and the operations that are valid for each type. For example, you cannot extract the square root of a word or write a number with a capital letter. One of the reasons why this recognition is easy to make is that words, numbers, and other designations look different. However, for a computer, all data types are ultimately reduced to a sequence of bits, so the difference in types should be made explicit.

Data types accepted in programming languages include natural and integer numbers, real (real) numbers (in the form of approximate decimals), letters, strings, etc.

In some programming languages, the type of each constant or variable is determined by the compiler by writing the assigned value; the presence of a decimal point, for example, can be a sign of a real number. Other languages require the programmer to explicitly specify the type of each variable, and this has one important advantage. Although the value of a variable can change many times during program execution, its type should never change; this means that the compiler can check the operations performed on this variable and make sure that all of them are consistent with the description of the variable type. Such a review can be carried out by analysing the entire text of the programme, in which case it will cover all possible actions defined by the programme.

Depending on the purpose of the programming language, the type protection provided at compile time can be more or less stringent. For example, THE Pascal language, which was originally primarily a tool for illustrating data structures and algorithms, retains very strict type protection from its original purpose. In most cases, the PASCAL compiler considers mixing data of different types in one expression or applying operations that are not typical for the data type as a fatal error. In contrast, C, which is primarily intended for system programming, is a language with very little type protection. C compilers only issue warnings in such cases. The lack of strict type protection gives the system programmer, who develops the program in the C language, additional opportunities, but such

programmer is responsible for the correctness of his actions.

The data structure is essentially a "spatial" concept: it can be reduced to a scheme of organization of information in computer memory. The algorithm is an appropriate procedural element in the structure of the program - it serves as a calculation recipe.

The first algorithms were invented to solve numerical problems such as multiplying numbers, finding the greatest common divisor, calculating trigonometric functions and others. Today, uncounted algorithms are equally important: they are designed for such tasks as, for example, searching for a given word in the text, planning events, sorting data in a specified order, etc. Uncounted algorithms operate with data that are not necessarily numbers: moreover, no deep mathematical concepts are needed to construct or understand them. This, however, does not mean that there is no place for mathematics in the study of such algorithms: on the contrary, accurate, mathematical methods are necessary in finding the best solutions to uncounted problems in proving the correctness of these solutions.

Data structures used in algorithms can be extremely complex. As a result, choosing the right data representation is often the key to successful programming and can have a greater impact on program performance than the details of the algorithm used. It is unlikely that there will ever be a General theory of the choice of data structures. The best thing you can do is to understand all the basic "bricks" and the structures assembled from them. The ability to apply this knowledge to the design of large systems is primarily a matter of engineering skill and practice.

2.2 Information and its representation in memory

Starting the study of data structures or information structures, it is necessary to clearly establish what is meant by information, how information is transmitted and how it is physically placed in the memory of the computer. The nature of the information We can say that the solution of each problem with the help of a computer involves recording information into memory, extracting information from memory and manipulating information. Can information be measured? In the theoretical and informational sense, information is considered as a measure of uncertainty resolution. Suppose that there are n possible States of some system

in which each state has a probability of p, and all probabilities are independent. Then the uncertainty of this system is defined as

$$H = -\sum(p(i) \cdot log_2(p(i)))$$

To measure the uncertainty of the system, a special unit called a bit is selected. A bit is a measure of the uncertainty (or information) associated with having only two possible States, such as true-false or Yes-no. The bit is used to measure both uncertainty and information. This is understandable, since the amount of information received is equal to the amount of uncertainty removed as a result of the information received.

In digital computers, there are three main types of storage devices: ultra-fast, RAM and external memory. Typically, over-memory is built on registers. Registers are used to temporarily store and transform information.

Some of the most important registers are contained in the computer's CPU. The CPU contains registers (sometimes called accumulators) into which arguments (i.e. operands) of arithmetic operations are placed. Addition, subtraction, multiplication and division of the information entered in the batteries is performed using very complex logic circuits. In addition, individual bits may be analyzed in batteries to verify that the normal sequence of control transmissions needs to be changed. In addition to storing operands and the results of arithmetic operations, registers are also used to temporarily store program commands and control information about the number of the next executed command.

RAM is designed to remember more constant in nature information. The most important property of RAM is addressability. This means that each memory cell has its own identifier that uniquely identifies it in the total array of memory cells. This identifier is called an address. The cell addresses are operands of those machine instructions that access memory. In the vast majority of modern computing systems, the unit of addressing is a byte cell consisting of 8 binary digits. A specific memory location or set of locations can be associated with a specific variable in a program. However, to perform arithmetic calculations in which a variable is involved, the value of the variable must be transferred from the memory location to the register before the calculations begin. If the result of the calculation is to be assigned to a variable, the resulting value must again be transferred from the corresponding register to the associated memory location[5].

During the execution of the program, its commands and data are mainly placed in memory cells. The complete set of elements of RAM is often called the main memory.

External memory serves primarily for long-term data storage. Characteristic of the data on the external memory is that they can be stored there even after the completion of the program that created them and can be subsequently reused by the same program when it is run again or by other programs. External memory is also used to store the programs themselves when they are not running. Since the cost of external memory is much less than the RAM, and the amount is much larger, another purpose of external memory - temporary storage of those codes and data of the running program that are not used at this stage of its execution. The active codes of the executed program and the data processed by it at this stage must necessarily be placed in the RAM, since direct exchange between external memory and operating devices (registers) is impossible.

As a data warehouse, external memory has basically the same properties as the RAM, including the property of addressability. Therefore, in principle, the data structures on the external memory can be the same as in the operational memory, and the algorithms for their processing can be the same. But external memory has a completely different physical nature, different access methods are used for it (at the physical level), and this access has different time characteristics. This leads to the fact that the structures and algorithms that are effective for RAM are not those for external memory.

Number system

To provide an appropriate basis for the study of data structures, existing types of number systems should be discussed: positional and non-positional.

Numbers are used to symbolize the number of objects. A very simple method of representing quantity is to use the same icons. In such a system, a one-to-one correspondence is established between the icons and the recalculated objects. For example, six objects can be represented as ****** or 111111. Such a system becomes very inconvenient if you try to use it to represent large quantities.

The Roman numeral system provides a partial solution to the problem of representation of a large number of objects. In the Roman system, additional characters are used to represent groups of icons. For example, you can assume that I=*, Y=IIII, X=YY, L=XXXXX, etc. the Specified value is represented

by combining characters according to a number of rules that depend to some extent on the position of the character in the number. The disadvantage of the system, which from the very beginning is based on the grouping of a certain set of characters in order to form a new character, is the fact that to represent very large quantities requires a lot of unique characters.

Positional number systems The positional number system uses a finite number R of unique characters. The value R is often called the base of the number system. In a positional system, the number is represented both by the symbols themselves and by their position in the number record. The number system with the base ten, or decimal system is positional. Consider, for example, the number 1303. It can be represented as:

$$1 * 10^3 + 3 * 10^2 + 0 * 10^1 + 3 * 10^0$$

(Hereafter, the symbol is used as a sign of the exponentiation operation.) Fractional numbers can also be represented in the positional system. For example, one fourth is written as 0.25, which is interpreted as:

$$2 * 10^(- 1) + 5 * 10^(- 2)$$

Another example of a positional number system is a binary system. The binary number 11001.101 represents the same number as the decimal number 26.625. The decomposition of a given binary number according to its positional representation is as follows:

$$1 * 2^4 + 1 * 2^3 + 0 * 2^1 + 1 * 2^0 + 1 * 2^(- 1) + 0 * 2^(- 2) + 1 * 2^(- 3) =$$
$$16 + 8 + 1 + 0.5 + 0.125 = 26.625.$$

The most common number systems have a base of 2.10.10 and 16, which are usually called binary, octal, decimal and hexadecimal systems, respectively. All computer technology works in binary notation, since the basic elements of computer technology have two stable States. Octal and hexadecimal systems are used for the convenience of working with large binary numbers. .Image of numbers in positional notation The representation of numbers in any positional number system with a natural base K (K -1) is based on their representation as a product of the integer power m of the base K on a polynomial from this base:

$$A_r = R^m \sum(a[i]R^{-i})$$

where: $a[i] = 0, 1, \ldots, R-1$ - digits of the R-th number system : n - number of digits (bit) used to represent the number: R - base of number system: m .... -2, -1. 0, -1, -2.... - order of number: R^{-i} is the positional weight of the i - th digit. Thus. in the decimal (R=10) system to represent the numbers used digits a = 0,1....9: in binary (R=2) - a = 0,1. in hexadecimal (R=16). $a = 0, 1....9, A, B, C, D, E, F$ where are capital letters A..F are equivalent to 10. respectively..15 in decimal system. For example

1. $815 = 10^3 * (8 * 10^(- 1) + 1 * 10^(- 2) + 5 * 10(-3)) = 8 * 10^2 + 1 * 10^1 + 5 * 10^0$:
2. $8.15 = 10^1 * (8 * 10^(- 1) + 1 * 10^(- 2) + 5 * 10^(- 3)) = 8 * 10^0 + 1 * 10^(- 1) + 5 * 10^(- 2)$:
3. $0.0815 = 10^(- 1) * (8 * 10^(- 1) + 1 * 10^(- 2) + 5 * 10^(- 3)) = 8 * 10^(- 2) + 1 * 10^(- 3) + 5 * 10^(- 4)$:

Converting numbers from one number system to another

When translating an integer (an integer part of a number) from one number system to another, the initial number (or an integer part) must be divided on the basis of the number system into which the translation is performed. Perform division until the quotient is less than the base of the new number system. The translation result is determined by the remainder of the division: the first remainder gives the lowest digit of the resulting number, the last quotient of the division gives the highest digit. When transferring the correct fraction from one number system to another number system, the fraction should be multiplied by the base of the number system to which the translation is performed. The integer part obtained after the first multiplication is the highest digit of the resulting number. Multiplication lead until the product becomes zero or the required number of characters after the dividing point is obtained[3]. For example.

1. convert the fractional number 0.243 from decimal to binary.
2. translate 164 integers from decimal number system to binary system.

When translating mixed numbers, the integer and fractional parts of the number are translated separately.

2.3 Classification of data structures

It is now possible to give a more specific definition of the machine-level presentation of information.

Regardless of the content and complexity of any data in the computer memory represented by a sequence of binary bits. or bits. and their values are the corresponding binary numbers. The data. considered as a sequence of bits. has a very simple organization or. in other words, poorly structured. For a person to describe and explore any complex data in terms of sequences of bits very uncomfortable. Larger and more meaningful than the bit, "building blocks" for the organization of arbitrary data are obtained on the basis of the concept of "structure of the given".

Under the DATA STRUCTURE in General. understand the many data elements and many relationships between them. This definition covers all possible approaches to data structuring. but each task uses one or another of its aspects. Therefore. an additional classification of data structures is introduced. the directions of which correspond to different aspects of their consideration. Before proceeding to the study of specific data structures. we give their General classification on several grounds. The concept of "PHYSICAL data structure" reflects the way the physical representation of data in the memory of the machine and is also called the storage structure. internal structure or memory structure.

Considering a data structure without considering its representation in machine memory is called an abstract or LOGICAL structure. In General, there is a difference between the logical and the corresponding physical structures, the degree of which depends on the structure itself and the characteristics of the environment in which it should be reflected. Because of this difference, there are procedures that display the logical structure in the physical and. conversely, the physical structure in the logical. These procedures also provide access to and perform various operations on physical structures, each operation being considered in relation to a logical or physical data structure.

There are SIMPLE (basic, primitive) data structures (types) and INTEGRATED (structured. composite, complex). Simple data structures are those that cannot be subdivided into parts larger than bits. From the point of view of physical structure. it is important that in this machine architecture, in this programming

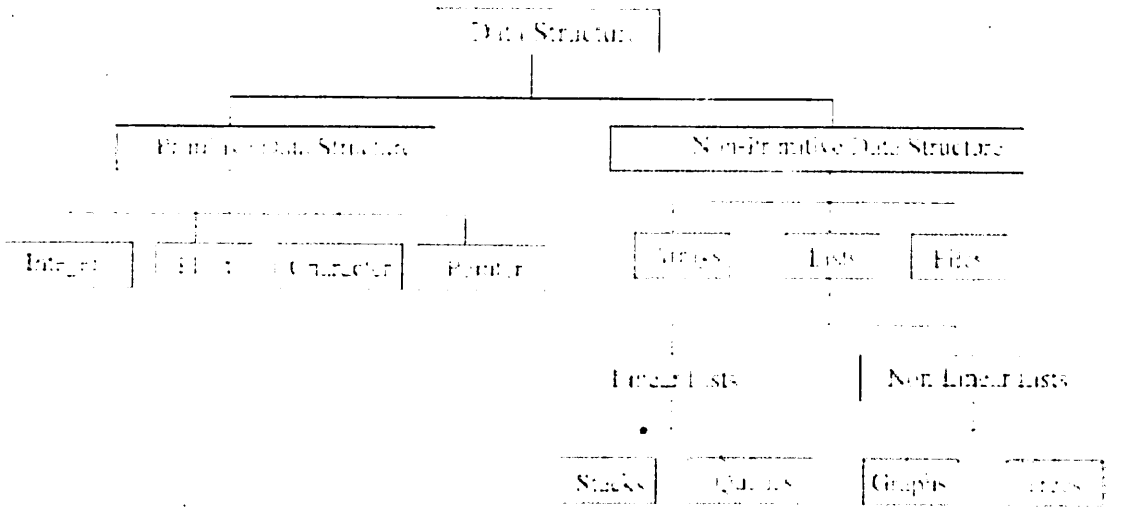


Figure 2.1: Basic data structures

system. we can always say in advance what will be the size of this simple type and what is the structure of its placement in memory. From a logical point of view. simple data are indivisible units. Integrated data structures are those whose components are other data structures - simple or, in turn, integrated. Integrated data structures are constructed by the programmer using data integration tools provided by programming languages.

Depending on the absence or presence of explicit relationships between data items. you should distinguish between DISJOINT structures (vectors, arrays, rows, stacks, queues) and CONNECTED structures (linked lists).

A very important feature of the data structure is its variability - the change in the number of elements and (or) relationships between the elements of the structure. The definition of structure variability does not reflect the fact that the values of data elements have changed, because in this case all data structures would have the property of variability. On the basis of variability distinguish the structure of STATIC, semi-static, DYNAMIC. Classification of data structures on the basis of variability is shown in Fig. 2.1. Basic data structures, static, semi-static and dynamic, are typical of RAM and are often referred to as operational structures. File structures correspond to data structures for external memory.

An important feature of the data structure is the ordering of its elements. On this basis the structure can be divided into LINEAR AND NONLINEAR structures.

Depending on the nature of the relative arrangement of elements in memory, linear structures can be divided into structures with a SEQUENTIAL distribution of elements in memory (vectors, strings, arrays, stacks, queues) and structures with an ARBITRARY CONNECTED distribution of elements in memory (simply connected, doubly connected lists). An example of nonlinear structures is multi-linked lists, trees, graphs.

In programming languages, the concept of "data structures" is closely related to the concept of "data types". Any data, i.e. constants, variables, function values or expressions, are characterized by their types.

Information for each type uniquely identifies:

1. the structure of data storage of the specified type, i.e. allocation of memory and representation of data in it, on the one hand, and interpretation of binary representation, on the other;

2. the set of valid values that can have one or another object of the described type;

3. a set of valid operations that are applicable to the object of the described type.

Operations on data structures Four General operations can be performed on any data structures: create, destroy, select (access), update.

The create operation is to allocate memory for the data structure. Memory can be allocated during program execution or at compile time. In a number of languages (for example, in C) for structured data constructed by the programmer, the creation operation also involves setting the initial values of the parameters created by the structure. For example, in PL/1, DECLARE N FIXED DECIMAL will allocate address space for the variable N. FORTRAN (Integer I), PASCAL (I: integer), and C (int I) will allocate memory for the corresponding variables as a result of the type description. For data structures declared in the program, memory is allocated automatically by means of programming systems either at the compilation stage or when the procedural block in which the corresponding variables are declared is activated. The programmer can allocate memory for data structures himself, using the procedures/functions of memory allocation/release available in the programming system. In object-oriented programming languages,

when developing a new object, procedures for creating and destroying it must be defined.

The main thing is that regardless of the programming language used, the data structures available in the program do not appear "from nothing", but are explicitly or implicitly declared by the operators of creating structures. As a result, all instances of structures in the program are allocated memory to accommodate them.

The operation of destroying data structures is the opposite of the creation operation. Some languages, such as BASIC, FORTRAN, do not allow the programmer to destroy the created data structures. The languages PL 1, C, PASCAL data structures available inside the unit are destroyed in the process of execution of the program when you exit that block. The destroy operation helps to use memory efficiently.

The select operation is used by programmers to access data within the structure itself. The form of the access operation depends on the type of data structure that is accessed. The accessor method is one of the most important properties of structures, especially since this property is directly related to the choice of a specific data structure[1].

The update operation allows you to change the data values in the data structure. An example of an update operation is an assignment operation, or a more complex form of parameter passing.

The above four operations are mandatory for all data structures and types. In addition to these General operations, specific operations that work only with data of this type (this structure) can be defined for each data structure. Specific operations are considered when considering each specific data structure.

2.4 Data structure and programming technology

Most of the authors of publications devoted to the structures and data organization, focus on the fact that the knowledge of the data structure to organize their storage and processing as efficiently as possible - from the point of view of minimizing cost of both memory and CPU time. Another, and perhaps more important, advantage of the structural approach to data is the ability to structure a complex software product. Modern industrial software packages are extremely complex products, the volume of which is estimated in thousands and millions of lines of code, and the complexity of development - hundreds of man-years. Naturally, it is impossible to develop such a software product "all at once", it should be presented in the form of some structure - components and links between them. The correct structuring of the product makes it possible at each stage of development to focus the developer's attention on one visible part of the product or to entrust the implementation of different parts to different performers.

When structuring large software products, an approach based on the structuring of algorithms, known as "top-down" design or "top-down programming", or an approach based on data structuring, known as "bottom-up" design or "bottom-up programming", may be applied.

In the first case, first of all, the actions that the program should perform are structured. A large and complex task facing the designed software product is presented in the form of several subtasks of a smaller volume. Thus, the module of the highest level, which is responsible for the solution of the whole problem, turns out to be quite simple and provides only a sequence of calls to the modules that implement the subtasks. At the first stage of design the modules of subtasks are executed in the form of "plugs". Then each subproblem is decomposed according to the same rules. The process of splitting into subtasks continues until at the next level of decomposition a subtask is obtained, the implementation of which will be quite foreseeable. In the limiting case, the decomposition can be brought to the point that the subtasks of the lowest level can be solved by elementary tools (for example, one operator of the selected programming language).

Another approach to structuring is based on data. A programmer who wants his program to have a real application in some application field should not forget that programming is data processing. In programs, you can invent as much intri-

cate and sophisticated algorithms as you want, but a real software product always has a Customer. The Customer has the input data, and he wants the output data to be obtained, and by what means it is provided - he is not interested. Thus, the task of any software product is to convert input data into output data. Programming tools provide a set of basic (simple, primitive) data types and operations on them. By integrating basic types, the programmer creates more complex data types and defines new operations on complex types. Here we can draw an analogy with the construction work: the basic types - "bricks", from which complex types are created - "building blocks". The "building blocks" compositions obtained in the first step are used as a basic set for the next step, which will result in even more complex data constructions and even more powerful operations on them, etc. Ideally, the last step of the composition gives data types corresponding to the input and output data of the task, and operations on these types implement the full task of the project.

Programmers who have a superficial understanding of structural programming often oppose top-down design to bottom-up design, adhering to one approach they have chosen. The implementation of any real project is always carried out in opposite ways, and, with constant correction of algorithm structures based on the results of the development of data structures.

Another extremely productive technological technique associated with data structuring is encapsulation. Its meaning is that the constructed new type of data - the "building block" - is designed in such a way that its internal structure becomes inaccessible to the programmer - user of this type. A programmer using this type of data in his program (in a higher-level module) can operate on this type of data only through calls to procedures defined for this type. The new data type is represented for it in the form of a "black box" for which inputs and outputs are known, but the content is unknown and unavailable.

Encapsulation is extremely useful both as a means of overcoming complexity and as a means of protecting against errors. The first goal is achieved due to the fact that the complexity of the internal structure of a new type of data and algorithms for performing operations on it is excluded from the field of view of the programmer-user. The second goal is achieved by the fact that the user's access is limited only by the obviously correct input points, therefore, the probability of errors is reduced.

Modern programming languages of block type (PASCAL, C) have sufficiently developed capabilities to build programs with a modular structure and control module access to data and procedures. Extensions languages additional features design types and their encapsulations makes a language object-oriented. Constructed and fully enclosed data types are objects, and procedures that work with their internal structure are methods of working with objects. At the same time, the very concept of programming is changing to a large extent. A programmer operating on objects tells the program WHAT to do with the object, not HOW to do it.

Database technology has evolved in parallel with, and not always in concert with, programming language technology. Partly this, and partly objective differences in the nature of the problems solved by database management systems (DBMS) and programming systems, caused some terminological and conceptual differences in the approach to data in these two areas. The key concept in DBMS is the concept of data model, basically identical to the concept of logical data structure. Note that the physical structure of the data in the DBMS is not considered at all. But the DBMS are software packages that display the physical structure in the logical (in the data model). To implement these packages, certain programming systems are used, DBMS developers, therefore, deal with data structures in terms of programming systems. For the user, the internal structure of the database and the physical structure of the data is completely transparent; it deals only with the data model and other concepts of the logical level.

3. Modeling systems

3.1 Simulation modeling system

An important class of systems to support strategic decision-making solutions. make up simulation systems. They allow you to create computer models that describe the various processes as they would have passed in reality. Such models can be "scrolled" in time many times. practicing various management decisions (analysis of "what–if") or simply accumulating the resulting data for the purpose of further statistical processing. Until recently. simulation modeling (IM) was perceived only as the tool of the scientist-researcher, which serves to identify the General patterns of the studied phenomena. Currently. THEY began to pay attention to managers (business analysts, consultants). considering it as a tool forecast. analysis and optimization (Borschev. 2008). There were works (including Russian-speaking). concerning the theme of simulation modeling for strategic decision-making.

For example. the monograph (Bulycheva. Miloradov, Khalikov. 2009) devoted to the modeling of market strategy of the enterprise. If at the stage of analysis and forecasting of the system THEY are helped to answer the questions "Why the company is developing anyway?"and "Where will the business move with time?"then at the synthesis stage they should help to answer the question "How to act to bring the company to prosperity?» Simulation models are used not only for activity analysis enterprises, but also to predict the behavior of the environment. Thus. the sub-step 2.2 will help THEM to determine the degree of attractiveness of selected areas of management under different scenarios. To do this, you need to design models. reflecting the major actors in the area of market forces: suppliers. customers. competitors, substitute products. new traders of market. Already now there are interesting examples of such studies, for example,

specialists of the company XJ Technologies has built a number of models dedicated to competitive analysis. Simulation experiments will also be useful directly in the selection of strategic decisions (sub-step 3.2). For example, you can use the supply chain model (on the web site) to make an informed decision about the strategy for building resource stocks. Finally, its application will increase the efficiency of planning the implementation of specific actions and resources within the framework of the chosen strategy (sub-stage 3.3). Today, three modeling paradigms are the most popular: system-dynamic (processes are described in the form of flowcharts), discrete-event (processes are described by a set of entities that generate individual events in time and react to them) and agent (the system is modeled multiple independent, but interacting with each other objects). In the work (Tenchurin, Shatilov, Avdoshin, 2009) devoted to the development of the information system of scenario strategic planning, the appropriateness of the use of system-dynamic methodology in CS. In (Barnabe, 2011) shows how you can use the system-dynamic model of the enterprise together with a system of key performance indicators. Job (Hell, Vidačić, Garača, 2009) is devoted to the mathematical model used for finding the allocation of resources that maximizes the achievement of the goals of the company. The model itself is static, but its use in the framework of discrete-event simulation model of the enterprise allows to develop an effective resource management strategy for a given period of time.

In (Druckenmiller, Acar, Trout, 2004), agent-based modeling is used in conjunction with cognitive maps to develop scenarios for the behavior of the passenger air transport market and make decisions about pricing policy airlines. In (Krasnoselsky, 2009) agent-based modeling is used for modeling of cellular communication market in with the aim of supporting decision-making concerning the integration strategies of companies operating in this market.

Finally, the article (Borschev, 2008) States that all three methodologies can successfully used to design models to support adoption strategic resource decisions maximizing the level of achievement of enterprise goals. The model itself is static, but its use in the framework of discrete-event simulation model of the enterprise allows to develop an effective resource management strategy for a given period of time.

Product of domestic company XJ Technologies AnyLogic it has the advantage of supporting all three simulation paradigms. The monograph (Karpov) is devoted

to modeling in AnyLogic, 2006). A separate paragraph of this work is devoted to solving problems related to strategic management of the organization.

The other three products support only the system-dynamic paradigm, but each of them has its advantages. So, Powersim is easy to integrate with external development environments. Based on created in this the system of models using additional programming tools was a prototype of the Center for situational analysis and forecasting of CEMI RAS was built. The center is a library of econometric models, which can be worked with by means of web-browser-based solutions of courses, maximizing the level of achievement of enterprise goals. The model itself is static, but its use in the framework of discrete-event simulation model of the enterprise allows to develop an effective resource management strategy for a given period of time. Simulation tools can be built into the System, for example, in Microsoft Office Excel there is a function "Analysis "chtoesli", which allows to evaluate the effects of different management alternatives. As a inhibiting factor in the widespread use of simulation in the management of enterprises in the article there is a shortage of qualified specialists. It is caused by the high school's isolation from solving real business problems. In addition, it should to recognize that creating an adequate simulation model to support strategic decision-making is a much more complex task than creating models for optimization of continuous processes at the enterprise. It due to the fact that the current processes, as a rule, amenable to formal description: each of them can be assigned a number of numerical parameters and fix their dynamics over a sufficient number of periods in order to further hypothesize, analyze, model and optimize the process. The formation of the strategy is aimed at developing solutions that entail a significant change in a certain set of characteristics of the enterprise (for example, update the range of products, expansion/reduction of scope of integration, the transition from one management model to another stocks). In addition, strategic decisions often lead to qualitative changes in the enterprise and may affect areas such as cultural, institutional, cognitive environment, type of staff, management structure etc the Problem situation in this it is difficult (and sometimes impossible) to describe the case in terms of a quantitative model and, ultimately, a programme, establet a library of econometric models, which can be implemented through web-browserdevices reginiussen that maximizes the level of achievement of the goals of the company. The model itself is static, but its use in the framework of discrete-

event simulation model of the enterprise allows to develop an effective resource management strategy for a given period of time.

System of expert estimation

If decision-making procedures in operational management can be automated (for example, many ERP-systems automatically balance the allocation of resources of the enterprise), the formation of the strategy is almost impossible to fully standardize, and therefore automate. This is largely due to the fact that in the strategic management of a greater role played by the subjective views of the decision maker.

Based on their knowledge, LPR can provide forecasts and solve difficult tasks for which the methods of management accounting and economic analysis are not applicable. Expert evaluation systems are designed to support decision-making based on subjective assessments.

The work is devoted to the application of expert assessments for strategic planning tasks. The risk of making the wrong decision the only LPR, rarely having all the necessary information and clearly aware of the trends in the development of the enterprise and the environment, can be reduced by attracting a variety of experts. In this case, the support systems for collective decision-making, forming a separate class of systems, closely related to the systems of expert evaluation, come to the aid. The paper describes the conceptual model of the ARgo system, which serves to support collective decision-making in the process of CS.

Expert evaluation systems can be used: in assessing the elements of the potential of the enterprise, ranking of economic zones according to the degree of required attractiveness (expert evaluation is almost the only way to determine the impact on demand in that or other zone of political, social, technological, environmental and other difficult to formalize factors), development of criteria for evaluating alternatives and criteria for reviewing the strategy, assessing the compliance of the desired and actual potential , ranking the formed variants of the integrated strategy, etc.

Examples of expert evaluation and support information systems collective decision-making. Expert decision support system (№ 1 in table. 13) contains a knowledge base – a set of rules for selecting appropriate support methods decision-making depending on the conditions characterizing a specific problem situation (Informatization of economic decision-making, 2009). In it solutions of Creative

Decisions and Expert Choice used the analytic hierarchy and analytic network. Package Expert Choice successfully use the following when planning their activities Forty business giants like General Motors. Lockheed. Ford Motor Company. Ferrari. General Electric and many others.

Expert system

For the purpose of the analysis of a condition of the enterprise. and also development of recommendations expert systems can be used to make strategic decisions (POWER STATION.) The main element of the ES is a knowledge base containing formalized rules (for example, "if–then" bundles), according to which the expert develops his decision. Thus. the subjective opinion of an expert (many experts) is embedded inside the system. Using some mechanism applying these rules (the inference scheme), ES argues like a man that leads her to answer the question.

Interest in ES, serving to assist in the development of management decisions, arose among developers and researchers for a long time. So I go to work describes Finex systems (research project) and ICS (eng. Integrated Consulting System developed by SRI International) created in the 1980s. First with the success for financial analysis, the second – for the analysis of competitive opportunities and planning of production strategy in the industry. characterized by a high degree of differentiation products'. Variants of architecture of modern expert system of strategic planning is described in the papers (Azadeh, Sharifi, Saberi. 2009; Huang, 2009). The paper (Subramoniam. Krishnankutty. 2002) describes the expert system. allows you to choose a method of strategic planning, depending on the capabilities and requirements of the enterprise.

Examples of ready-made it solutions. which are expert systems are given. According to the generally accepted classification of ES by the solved the problem presented in the monograph. given in the table of the system can be attributed to the class "Diagnostic ES", as well as to class "S supporting decision-making."

It is obvious that managers. analysts and consultants who develop an enterprise strategy need to know the methods of strategic management and have an idea of the state of Affairs in related areas (Economics. law. technology. marketing). In addition. ideally they should be aware of hidden trends in the development of the industry and patterns of demand for products , to share experiences in the adoption of successful solutions. quickly find sources of information. and.

finally, effectively conduct a dialogue on various issues of formation of the enterprise strategy. Anyway, all this associated with the creation/attraction, storage and distribution of knowledge, as well as ensuring access to them. According to (Makarov, Kleiner, 2007) under the knowledge of the organization should be understood as a generalized, systematic, past accepted in the organization of public expertise and related to the essential aspects of the organization. To help in the work with knowledge can knowledge management information systems. Because a wide range of supported processes, these systems are quite diverse (classification of knowledge management systems with examples is given, for example, in the article).

Forty three In (Zach, 2010) especially stands out is the concept of governance strategic knowledge, which is defined as the processes and infrastructure that a firm uses to obtain, create, and distribute knowledge required to formulate a strategy. As an example, the company, which was formed online-mechanism for collecting competitive knowledge about the external environment from employees engaged in sales and service of the goods, for subsequent transfer to persons formulating the strategy. Strategic knowledge management should be clearly separated from strategy knowledge management (cognitive strategy), which is one of the components of a comprehensive strategy of the enterprise and is a set of strategic decisions that determine the structure and functioning of knowledge of processes. In the book (Mintzberg, Ahlstrand, Lampel, 2000) management knowledge and strategic management are combined in a separate the direction of strategic thought – "schools of learning".

In work (Perangai, 2010) to formalize procedural knowledge in management are encouraged to use the technology of structuring algorithms decisions (SAVUR-technology). In its basis is the visual programming language DRAKON, aims to improve the efficiency of the process of avtotormozami knowledge (that is, independent of the process of formalization of their knowledge). A large number of DRAGON-diagrams from the region strategic planning is presented in the paper (Pavlova, 2002). In the management of strategic knowledge, first of all, you can use the following subclasses of systems of knowledge management as a portal solution, system, enterprise content management, knowledge bases, and distributed advisory network.

Portal solutions allow you to create an internal web-site of the enterprise, with

which you can publish news and other messages for employees, provide access to files and documents. Often corporate portals have functions to support collaboration, providing individual projects or divisions of the organization with virtual workspaces with task management functions, individual and group calendar, etc. the Most developed portal solutions allow you to organize access not only to data, but also to applications.

The presence of a corporate portal in the company will be useful in practice at all stages of formation of complex strategy of the enterprise. Significantly it will be easier to inform employees about the beginning of the strate formation project- Forty four GII. It will be easier to schedule meetings with stakeholders (sub-step 1.1), conduct a survey of employees, provide them with access to working documents and reports (sub-step 1.2, stages 2, 3, 4). The portal can become a platform for publishing a simplified version of the strategy and thus everyone with access by the portal, the employee will be informed of the decisions taken (sub-step 4.2). You can also use the portal as part of a compliance mechanism decisions made at the enterprise of the developed strategy (sub-step 4.3). For this should be published in a dedicated section describing all significant decisions with an explanation of how they correspond to the strategy. Opportunity open discussion of decisions by all stakeholders will allow for more effective strategy development. An example of the implementation of a corporate portal to simplify the formation of strategic plans is specifically devoted to the article (Milett, Togamau, Rhodes, Clarke, Carswell, 2005).

In comparison with portal solutions, corporate information management systems, as a rule, have additional features: document version control, identification of relationships between employees and documents (who creates, uses, approves, edits), search for experts on the basis of authorship. In the paper (Hedelin, Allwood, 2002), devoted to identifying the needs of Executive Directors in it solutions for SU, the majority of respondents noted the urgent need for tools for quick search and access to information sources (the need for search tools not only documents, but also employees with the necessary knowledge). Corporate information management systems are well suited for these tasks and, as well as corporate portals, can be effectively used at almost all stages of the procedure for the formation of the enterprise strategy.

Another important class of knowledge management systems are databases.

knowledges". Here, the term "knowledge base" does not mean a system of formalized rules, as in the theory of ES, and an electronic encyclopedia, formed by experts and equipped with advanced tools for classification and search for the necessary information. Especially useful is the use of knowledge bases in consulting firms, the main capital of which is knowledge in the field of enterprise management. For example, ErnstYoung uses a database containing more than 5.000 basic methods of organizing production processes implemented by more than in 30 countries (Telnov, 2004). A knowledge base of PricewaterhouseCoopers has an independent trademark Global Best Practices (№ 9 in table. 15, the present-Forty five access to the database for external users is limited). Sometimes portal solutions allow you to create knowledge bases using Wiki technology.

Since the knowledge base can reveal useful experience in almost all aspects of the formation of the strategy, their use is not limited to specific steps. It often happens that the Manager making a strategic decision, it is necessary to seek advice from colleagues or experts in the field of Economics, politics, law, management. After listening to the problem statement, they can suggest the right solution based on their experience and knowledge. As a rule, this knowledge is implicit – it can not be fixed on paper and formalized in the form of set of rules. Therefore, to "force" someone else's knowledge "to work for themselves», the Manager must have personal contact with the bearer of knowledge. For remote holding of such mini-consulting meetings there are such solutions as distributed Advisory networks (eng. Peer Advisory Network). The use of a distributed Advisory network will be particularly important useful at the stage of strategic analysis , in the formation of sketches strategies (sub-step 3.1), development of strategic options , and when making the resulting decision.

3.2 Recommendations and implementation

The main purpose of this publication was to highlight the existing market tools potentially useful in the formation of the enterprise strategy. This does not mean that the introduction of these systems is unreasonable will have a positive effect on the process of formation and implementation of the strategy at a particular enterprise. Each company is unique, and the decision to implement a particular information system should come from the information needs of the persons

responsible for the strategy (Executive Director, analysts, consultants).

At the same time, in order to facilitate the selection process of DSS for CS, this paper describes the universal procedures for the formation of a comprehensive strategy of the enterprise (see section 1). Each solution is associated with any (sub)class of DSS at SU, and each (sub)class, in turn, is associated with certain procedures for the formation of the enterprise strategy. Thus, an overall picture that reflects the potential It support for strategic decision-making. This section provides some guidance on which systems should be implemented in a particular case.

1. If the enterprise databases have accumulated a sufficiently large volume factual data (for example, sales and/or procurement data in various sections – by customer/supplier segments, departments, regions), and experts recognize the value of their in-depth analysis in the formation of strategies, you should think about implementing OLAP and Knowledge tools Discovery (you may need to pre-organize your data warehouse). You can start implementing such systems with SaaS solutions. This will help you understand, will they bring any benefit without spending significant funds.

2. The implementation of external information systems should be approached with caution. Since different information will be important for different enterprises, the selection of solutions should be carried out strictly individually. Herewith it is necessary to pay attention to whether potential sources of information have interfaces for integration with existing enterprise information systems

3. If the enterprise has an entrepreneurial type and is at the stage of formation (ideas), then "light" business planning tools will be useful. They incorporate techniques that can be used entrepreneurs who are not experienced in planning. As said in subsection 3.3, business planning refers to the level of tactics, however, well a well-thought business plan will form part of the strategic decisions according to the principle "bottom-up".

4. Risk management systems are particularly useful for companies working in the project style, as well as for financial companies. This is due to the fact that the strategy of project enterprises is largely determined a set of

executable projects (and Vice versa, a set of projects is formed on the basis of strategy), and, as is known from the theory of project management, for successful project implementation requires a risk management system. Financial companies initially "earn on risks" (Starinskaya, 2007).

5. Complex BPM solutions are recommended to be implemented in large companies. Small and medium-sized companies may not have the problems that this class of DSS is aimed at solving – the relationship between strategy and operational plans and motivations of performers in small and medium-sized companies, as a rule, are easily visible, and the resulting deviations are eliminated by the owners or a small team of managers. In large companies, these relationships are usually, they are quite complex because of the large number of data centers responsibilities and levels of coordination of decisions, therefore the use of integrated it systems, standardizing the planning and control processes, should improve their effectiveness (Lyamin, 2007). To bring simulation systems to the enterprise benefit, it should have highly skilled personnel familiar with methodology and tools. Although some authors claim that simulation can easily become a tool of a top manager, in Russia, the design of models and development of recommendations with their help remains the prerogative of specialized consulting companies.

6. Business modeling systems, as well as simulation systems, are designed for use by analysts who are familiar with the methodologies they support. If the company has a specialized analytical Department or an employee familiar with the basics of business modeling, such systems can be very useful – they will allow visualize the structure of the enterprise, business processes, which is very important when planning management and restructuring strategies.

7. As the development of the strategy ultimately boils down to estimating for the preference of the decision maker various options, the many stages of strategy formation will be useful to use a system of expert estimation. In (Gulyaev, 2001) explains the use of expert assessments in the ranking of potential areas of management, in the monograph (Kleiner, 2008) – when selection of the resulting version of the integrated strategy in the work – to make forecasts on the market share and value of the company, in the article (Carlucci, 2010) – in the selection of key performance indicators companies.

General principles of application of expert evaluation methods in strategic planning is considered in the work. First of all, you should pay attention to the expert evaluation systems provided on the basis of the SaaS model – their implementation will not require significant costs, and their use can be abandoned at any time.

8. The advice provided by expert systems is additional information for reflection, which will not prevent the adoption of responsible decisions'. Typically, ES have a user-friendly interface and simple logic of work. When you select the ES should take into account when it developed the system (the rules that worked well in the past may be out of date) and on the use of the companies they are designed.

9. If the enterprise has distributed character (a set of branches, divisions, points of coordination of decisions) or maintenance is implied strategic developments within the group (cluster) of enterprises, the use of portal solutions and corporate information management systems will increase the efficiency of the strategy formation process, at least from a technical point of view. Improvement will be achieved at the expense of convenient (at proper organization of the portal) access to reports and documents arising during the formation of the strategy, as well as through the possibility of organizing joint work with documents and remote management of meetings

Once again, we note that, introducing the DSS at a particular enterprise, it should be first, develop a procedure for the formation of the strategy (it may not be fully consistent with the proposed universal procedures in this work), and then choose the tools so that they correspond to this order, supporting certain stages of the strategy formation process and effectively interacting with each other. It is necessary to pay attention to the fact that many information systems incorporate a certain methodology, best practices, a view on how the strategy should be formed, what its elements are important (determining) and which are secondary (dependent). In this regard, in practice, it is often wiser to "adjust" the mechanism strategic management under the methodology used in a particular information system (with some modifications of this methodology), and not the information system under the existing mechanism of the SU. This "adjustment" may lead to the need to restructure the business processes of the enterprise using approaches

to managing organizational changes (ashmarina, Gerasimov, 2011). In this case, we will talk about the basis of the enterprise – processes managements. Thus, the implementation of the DSS may be a strategic issue, so it is imperative to assess the benefits of the application this or that system. Methods of such assessment for analytical level systems described in the article (Seredenko, 2010).

Real implementation projects DSS at SU are investigated using method of situational analysis (eng. case-study). In the reports of these studies typically, the benefits (costs) received (incurred) are detailed) the enterprise at implementation of the considered system. Also list difficulties encountered during the implementation of projects and steps that could be taken to avoid them. It follows that it will be useful if the specialists responsible for the DSS implementation project get acquainted with descriptions of the implementation of systems set out in the scientific literature. As examples of studies on the implementation of DSS in SU can be a report on implementation of Microsoft SharePoint portal solution to support strategic planning at the University (Milett, Togamau, Rhodes, Clarke, Carswell, 2005) and the work (Shen-Hsieh, Schindler, 2002) containing a detailed review project implementation of visual Analytics system to support the adoption of strategic decisions in a pharmaceutical company.

When implementing the DSS, it should also be borne in mind that the main task is not to provide the DSS with as much information as possible, but, on the contrary, providing a small amount of the most important data needed for decision-making. Today, information overload is nothing no less a problem than the lack of information (although it has been found in research (Hedelin, Allwood, 2002), Executive Directors, responsible for strategic decision-making, almost never experience information overload). Information systems, on the one hand, allow to cope with this overload (for example, corporate portals allow quickly search for the necessary documents, set up virtual offices, based on individual information needs of each user), but, on the other hand, are sources of unnecessary information. If the company a strategic planning division will be established, with sufficient staff, to be able to work effectively with a variety of information systems to support SU. In this case, one of the main tasks of such a Department will be "filtering" the information received in order to preparation of summary reports for the LPR. In a small company, it is enough to introduce 2-3 tools to get a useful effect and, at the same time, avoid "idle" systems that will be used very rarely. In conclusion

of this section we give some references to electronic directories of IP for business. where you get acquainted with the possibilities of different it solutions and to choose a system that is appropriate for specific tasks.

3.3 Mapreduce

Chapter 2 introduced the MapReduce model. This Chapter discusses practical aspects of developing MapReduce applications in Hadoop. MapReduce programs are written according to a certain scheme. First, you write display and convolution functions — ideally together with unit tests that check whether the functions work as expected. Then, you write a control program to run the job, which can be run from the integrated development environment (IDE) with a small subset of the data for functional test. If the startup fails, you use debugger your IDE to find the source of the error. With complete information, you Refine unit tests and display/convolution functions, by ensuring that these inputs are processed correctly. When the program will work with a small set of data as expected, you can "release" it into the cluster. It is likely that when you run the full set of data, you will find new problems that are corrected by the refinement of tests and functions display convolution. Debugging programs in a cluster is not an easy task, so we will look at some standard techniques that simplify it.

When the program is running, it is worth doing optimization — first, after some standard checks to speed up the work of MapReduce programs, then by profiling tasks. Profiling distributed programmer configuration... Two hundred three it also creates a lot of problems, but Hadoop provides entry points that simplify this process. Before you start writing MapReduce programs, you should prepare and set up the development environment. And for this it is necessary to understand how to Hadoop configuration is configured.

During the development of Hadoop applications developer frequently switches between running the application in local mode and in the cluster. What's more, you can work with multiple clusters or create a local pseudo-distributed cluster to be used for testing ("pseudo-distributed" is called a cluster, all daemons of which are running on the local computer.) For example, you can create hadoop configuration files with connection settings for each cluster in which you run the program, and specify the file when you run hadoop applications or tools. Such it is

desirable to store the files outside the hadoop installation directory tree. as 1 Karmasphere provides Eclipse and NetBeans plug-ins for developing and running jobs MapReduce jobs and browsing Hadoop clusters. Configuration management... Two hundred nine this makes it easy to switch between Hadoop versions without duplicating or loss of settings. We assume that there is a directory named conf that contains three configuration file: hadoop-local.xml. hadoop-localhost.xml and hadoop-cluster(xml files are included in the code examples in the book). Note that there is nothing special about file names: this is nothing more than a convenient scheme for packing configuration parameters. **Security** Early versions of Hadoop assumed that HDFS and MapReduce clusters would be used by groups of interacting users in a secure environment. Access restriction measures were designed to prevent accidental data loss. not unauthorized access to it. For example. the system permissions in HDFS will prevent accidental deletion of the entire file system due to an error in the program or a accidentally entered hadoop fs -rmr command. but not will prevent an attacker from impersonating a privileged user (see below). "Setting user ID", p. 210) for handling or deleting any data in the cluster. The system lacked a secure authentication mechanism that allowed would Hadoop make sure that the user wanting to perform the operation in the cluster, the one he claims to be, and he can be trusted. HDFS file permissions provide only an authorization mechanism, that is, define operations that a particular user can perform on a particular file. For example. only a specific file can be read a group of users. and no user who is not a member of that group can read the file. However, authorization alone is not enough because an attacker with network access to the cluster could gain access to the system by falsifying personal data. In practice, access to data.

This was the situation in 2009, when Yahoo! instructed a group of engineers to implement a secure authentication mechanism for Hadoop. System Hadoop itself does not manage user credentials; instead it uses Kerberos — a proven open-source network authentication Protocol. In turn. Kerberos does not control permissions. Kerberos only confirms that the user is who he or she claims to be: hadoop task — to determine whether the user is allowed to perform the specified actions. Kerberos is a fairly extensive topic. so here we consider only the use of Kerberos in the context of Hadoop. Readers who need for more information, we recommend that you refer to the book by Jason Garman (Jason Harman) "Kerberos: The

Definitive Guide».

Distributed systems — such as HDFS or MapReduce — involve multiple client-server interactions. each of which must be authenticated. For example. an HDFS read operation requires multiple accesses to a name node and one or more data nodes.

Since the use of three-phase Kerberos ticket exchange Protocol for authenticating each case in the loaded cluster will create a high load on KDC. Hadoop uses delegation tokens (delegation tokens) to authenticate subsequent requests without reconnecting to the server The KDC. Hadoop creates and uses delegation tokens transparently on behalf of the users. so the user will not be required to do anything other than run the kinit command to log in. Nevertheless, it is desirable at least in General it is hard to imagine how this mechanism works. The delegation token is generated by the server (the name node in our case). Their you can think of it as a shared secret between the client and the server. When RPC first accesses the name node. the client does not have a delegation token, therefore. the client uses Kerberos to perform authentication. As a part the response it receives is a delegation token from the name host. On subsequent calls. the client presents a delegation token that the name node can verify (because the token was generated using a private key); therefore. the client authenticates to the server.

When a client wants to perform an operation on HDFS blocks, it uses a special a type of delegation token is a so-called block access token that is passed by a name node to a client in response to a metadata request. Client uses a block access token when authenticating yourself on data nodes. This is only possible because the name node provides access to its private key used to generate the block access token, the data nodes. The key is transmitted through the periodic signal channel, and allows the data nodes to check the block access token. So, to the block HDFS can only be accessed by a client that has received a valid token block access from host names. This closes the security defectin Hadoop. when to gain access to the unit was a single block ID. To enable this mode. you must specify dfs.block.access.token.enable set to true.

In MapReduce, job resources and metadata (JAR files, input splits, and configuration files) are available in HDFS for the job tracker, and user code is executed on task trackers and accesses files in HDFS (this process is explained in the "Running MapReduce jobs"). Markers delegations are used by job and task trackers

to access HDFS in the course of the assignment. After a job is finished, the delegation tokens are invalidated. Delegation tokens are automatically obtained for the HDFS instance default. If your job needs to access other clusters HDFS, you can load delegation tokens for these clusters by specifying the mapreduce property.job.hdfs-servers comma separated list of HDFS URIs.

3.4 Description of the MapReduce model

Before starting the description of the MapReduce model itself, I would like to note that it has only an introductory function and contains a large number of simplifications. As already mentioned in the introduction, MapReduce programs are used to process large amounts of data, and therefore usually run on clusters of several nodes. The model is based on two functions Map and Reduce in General similar to those that are often used in functional programming.

All MapReduce programs consist of two stages:

1. At the Map stage, the input data is divided into parts and sent to nodes. On each node, the Map function is called, the result of which is a list of pairs: (intermediate key, intermediate value). Further, the data obtained using the partition function are distributed to the machines that will perform the subsequent processing. A little more on the partition function, as it will be largely key to this work. Partition is a mapping of the set of intermediate keys to the set $[0, k1]$, where k is the number of machines. In the classic version of MapReduce, the same intermediate keys must get on the same machine for correct processing, so this function should not have side effects.

2. In the Reduce stage, the intermediate data on each machine is sorted and grouped by the intermediate key, and then the reduce function is called on the intermediate key. The convolution function takes an intermediate key and a list of all intermediate values mapped to it, and returns some pair The advantage of this model is that the Map and Reduce stages can be produced in parallel on a variety of machines, that there are no arbitrary shared data and, as a consequence, constant network interaction required to synchronize data on the nodes. All this allows the system to be reliable and highly scalable. Figure 3.1 shows the main stages of MapReduce programs:

Problem

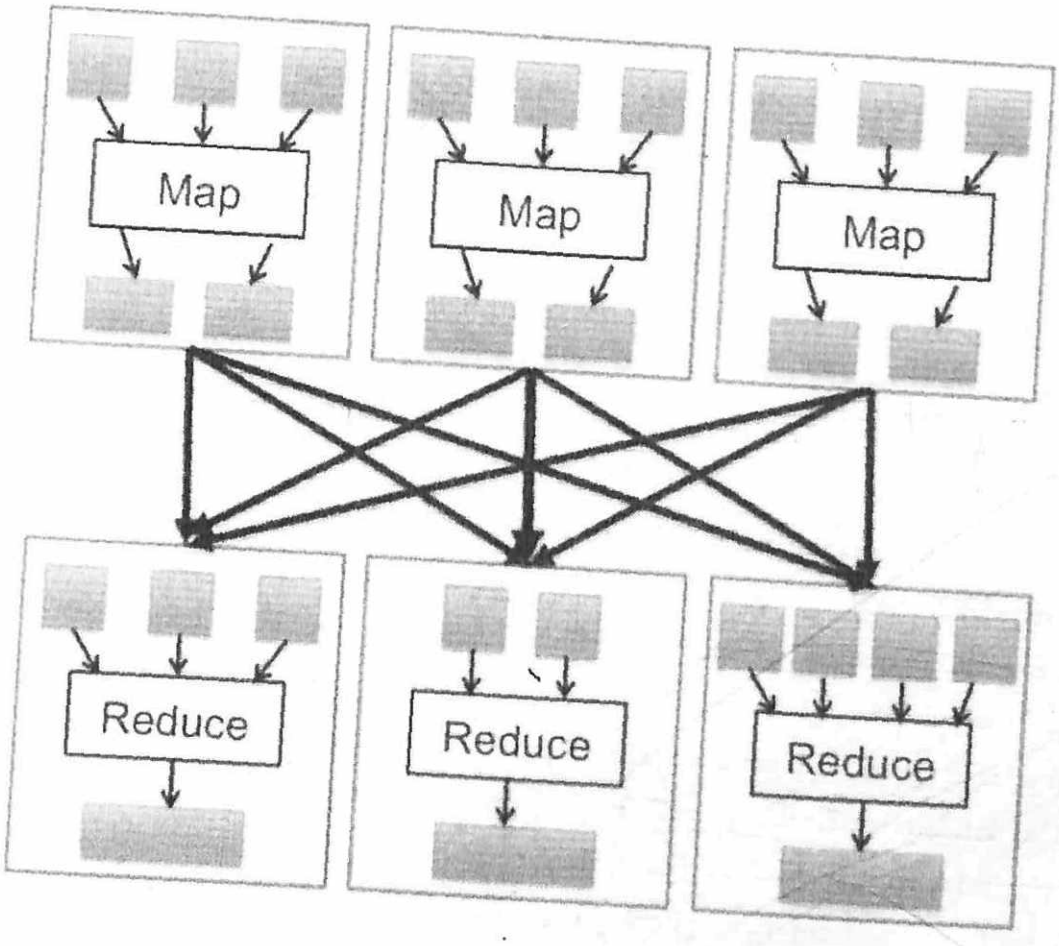


Figure 3.1: Model MapReduce

On some MapReduce tasks, you can observe the following situation: one or more Reduce tasks run much longer than others, preventing subsequent data processing from starting. It is logical to assume that this may be due to the fact that the volume of processed data on different reducers varies greatly, as a consequence of the unsuccessful distribution of intermediate keys.

Prerequisites

Real-world MapReduce tasks are performed with some periodicity from day to day on data that changes slightly. In view of this, it seems possible to use the statistics of distribution of intermediate keys and values obtained during previous runs in order to optimize the load on the reducers. A similar idea is used in relational databases when analyzing the cost of a query plan to predict the size of intermediate data.

Existing work on optimization In this part we will consider some works on optimization of MapReduce programs very briefly. In most of the optimization work, SQL queries are adapted to MapReduce. An example of such work can be the work [5], which presents the Manimal tool that solves the problem of finding projection, sampling, aggregation operations in MapReduce Programs that are mixed with other program logic, and applies optimizations for these operations. In another work [6] considered the optimization of the Join in MapReduce tasks. The strategies presented in it work well in the case of join-a single very large table with several small ones.

Open implementation of MapReduce.The Hadoop Project

The most common implementation of the MapReduce model is HadoopTM Project, the main contribution to which was made by the developers from Facebook and Yahoo! Hadoop has been successfully implemented in these companies. Also, this project is officially supported on Amazon EC2, where official AMI (Amazon Machine Images) are available, which immediately contain Hadoop. In addition, the Hadoop distribution package contains tools that make it relatively easy to deploy a cluster with Hadoop on Amazon EC2 and run MapReduce programs. Unfortunately, these tools and AMI have not been updated for a long time and are very outdated. In view of the broad support for this project and the ability to deploy the cluster on Amazon EC2, it was decided to conduct this study on its basis. Consider the subprojects that make up it:

1. Hadoop Common: common utility classes required by other Hadoop sub-

projects.

2. Hadoop Distributed File System (HDFSTM): a distributed file system with high bandwidth.

3. Hadoop MapReduce: a framework for writing and executing MapReduce programs on computing clusters.

The greatest interest for us represents the last sub-project of Hadoop MapReduce. We describe the key components included in it, and which will be used in the future in this work. The most important class for this study is Partitioner, an abstract class that is responsible for distributing keys obtained in the Map stage to Reduce machines. It has only one method with signature:

```
abstract class Partitioner<KEY, VAL> {
    abstract int getPartition(KEY key, VAL value, int k)
}
```

This method determines to which of the k machines the pair (key, value) will be sent for convolution. The default implementation of this class is the *HashPartitioner* class, in which the *getPartition* method takes the hash code of the key modulo k. It is worth noting that this implementation is not specific to the Hadoop project, it was mentioned in the very first publication about the MapReduce paradigm.

The other two base classes are the abstract classes Mapper and Reducer. Classes that inherit from Mapper implement a custom map function, just as the classes that inherit from Reducer implement a custom reduce function.

3.5 Problem statement

The aim of this work is to study the applicability and relevance of statistical optimization in MapReduce problems. This requires:

1. Implement collection of statistics of distribution of intermediate values by intermediate keys.

2. Implement an algorithm that provides a uniform load on the Reduce machine based on the collected statistics.

3. Implement tools for comparing distributions of intermediate keys to the reduce machines.

4. To carry out testing and comparative analysis of the optimized version on

real tasks.

Algorithm

Let's say that we are able to collect statistics on intermediate keys and on the basis of it to predict the number of intermediate values corresponding to each intermediate key or group of keys. With the help of this data it is necessary to load reduce machines so that the reduce stage on all of them takes the same time. We will proceed from the natural assumption that the execution time of the reduce task on one machine is proportional to the total number of intermediate values for all intermediate keys processed on this machine. Thus, I want to ensure that this number on all machines was the same. If we number the keys, we can get an array in which the number of intermediate values corresponding to the i intermediate key is on the i position, and this array needs to be divided into k as close as possible to the sum of the sets.

3.6 Mathematical formulation of the problem

Given an array of positive integers $a_1, \ldots, a_n$ and it must be divided into

k - sets $M_1, \ldots, M_k$ so that the difference is max $(S_1, \ldots, S_k)$ - min $(S_1, \ldots, S_k)$, was minimal where $S_k = \sum_{a \in Mk} a$

This is one of the special cases of the well-known K-balance partition problem, which in life occurs in different formulations such as: distribution of network requests to servers or tasks by processors; it is NP-complete. To solve this and similar problems with different restrictions on the input data, there are many research works, the purpose of this work was not to develop or improve existing approaches. The current implementation uses a simple algorithm similar to the first Fit Descending algorithm, which solves the "container packing problem".
Steps of this algorithm:

1. Let's sort the array in descending order a.
2. On i iteration is taken element a_i and placed in a lot with a minimal amount at the moment.

Architecture

In the previous section, it was assumed that we are able to collect statistics on the distribution of intermediate values by intermediate keys. This section will focus on the following questions: how it can be collected, which of the collec-

tion options has been implemented and what are the points of expansion in this implementation.

So, the first and fundamental question to answer is: what capabilities does the Hadoop API provide to collect statistics. The data required for statistics is available from the end of the Map stage to the beginning of the Reduce stage. Consider where you can add statistics collection in this interval:

1. The end stage Map, based on the class *Mapper*. The first way is to attempt to intercept the pair (intermediate key, intermediate value) while they are written to an instance of the *Mapperclass.Context*. For this you would need to create a class, a proxy class *Mapper.Context*, except for the fact that the method *write* (*KOUT key*, *VOUT value*) still would have updated statistics. The next step would be to create a new class that inherits *Mapper* and passes it to the method *map* (*KEYIN key*, *VALUEIN value*, *Mapper.Context context*), not the original context, but its proxy described above. This approach has the disadvantage that in order to try our optimization on already existing programs, it is necessary to make changes to the code, namely, to change the parent class for *Mapper*.

2. The end stage Map, based on the class *Partitioner*. This is the second way to collect statistics at the end of the Map stage. In this way is created an abstract class that are inherited from *Partitioner*, before calling a real challenge *getPartition*(*KEY key*, *VALUE value*, *int k*) will update statistics. Unfortunately, this method also contains its pitfalls: *Partitioner* does not have life cycle methods. This can be bad if the statistics are just saved to a file, as we need to know when *Partitioner* you have finished your work, and you can write the collected statistics to the file system.

3. The beginning of the Reduce phase, based on the class *Reducer*. The last option is to update the statistics in the descendant of the *Reducer* class before calling the *reduce* function. Let's consider its disadvantages: first, this option has the same disadvantage as the first option, second, additional iteration over all intermediate values is extremely dangerous, since in the case of a large number of disk operations will be performed, which can negate the optimization attempts.

To facilitate the further use of this development, it was decided to implement the collection of statistics on the basis of *Partitioner*. (It requires to register new *Partitioner* in the configuration files that would have to do anyway, since

we model and the algorithm of key distribution). The next step after the collection of statistics is its storage, and at this stage, too, there are two alternative ways:

1. store statistics in the file system;

2. store statistics in the database;

In view of the fact that the second way is much heavier, as it pulls new dependencies into the project and requires additional deployment of the database server in the cluster, and does not have the same obvious and significant advantages, it was decided to go the first and simplest way.

Once the choice is made on the basis of which components statistics will be collected, and where it will be stored, we will pay attention to the statistics itself. The obvious and unpleasant fact is that it is impossible to store it as a pair of intermediate key and the number of corresponding intermediate values, due to the fact that Map-Reduce processes huge amounts of data, and as a result:

1. there can be a lot of intermediate keys;

2. the keys themselves can be heavy, that is, take up a lot of disk space. For example, intermediate keys can be long strings;

Because of this problem, it is necessary to combine keys into groups, and distribute these groups of keys to reduce machines. And moving on to the software implementation of this work, this need was taken into account in the Statistical interface, which should be implemented by classes that collect and process statistics.

```
public interface Statistic<K, V, M> extends Writable {
    void add(K key, V value);
    Converter<K, M> getConverter();
    Map<M, Integer> export();
}

public interface Converter<K, M> {
    M convert(K key);
}
```

Statistic $< K, L, M >$correctly read as follows: statistics of the host keys K class, value class V, and uniting the key groups of class M. The standard implementation of this interface will be described below, but first we will briefly describe why we need some of its methods. Note that this interface inherits the *Writable* package *org.interface.apache.hadoop.io*, which contains the methods required to serialize objects of this class when saving it to disk or transferring it

over the network. Back to the interface methods *Statistic*:

1. *add* - adds to statistics *key* and *value*.

2. *getConverter* - returns a function that maps the key to the group that the key will belong to. This function should return the same group for the same key, otherwise the pairs with the same intermediate key may appear on different reduce machines. You should also keep in mind that this function will be called for each pair (intermediate key, intermediate value).

3. *export* – returns *Map*, in which each key group is mapped to the expected number of intermediate values mapped to all intermediate keys in that group.

The *HashStatistic* class is the default implementation of the Statistical interface. It groups keys by hash code taken modulo some number N, which greatly exceeds the number of machines. Based on the above algorithm and statistics interface, an abstract *StatistiscalPartitioner* class is created, which has one abstract method:

```
abstract Statistic<KEY, VALUE M> newStatistic();
```

Class *StatisticalPartitioner* with this method reads the statistics from the previous runs, it distributes the intermediate key. *HStatisticalPartitioner* is heirs *StatisticalPartitioner* using HashStatistic class as an implementation of the interface Statistic. *HStatisticalPartitioner* is the simplest example of a *Partitioner* that works based on the collected statistics of the distribution of intermediate values. Even such a naive implementation is enough to test the applicability of statistical optimization in the MapReduce paradigm.

The essence of MapReduce

The implementation of the algorithm takes on input 3 arguments:

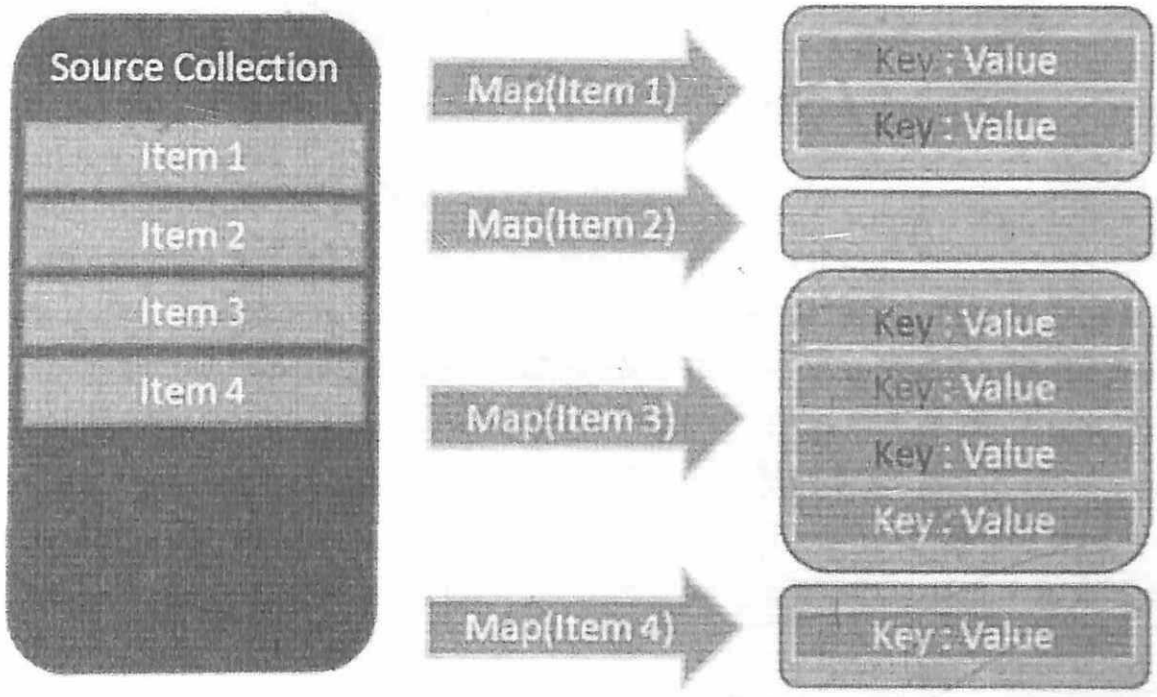
1. source collection
2. Map function
3. Reduce function
4. returns the new data collection after processing.

```
Collection MapReduce(Collection source, Function map, Function reduce)
```

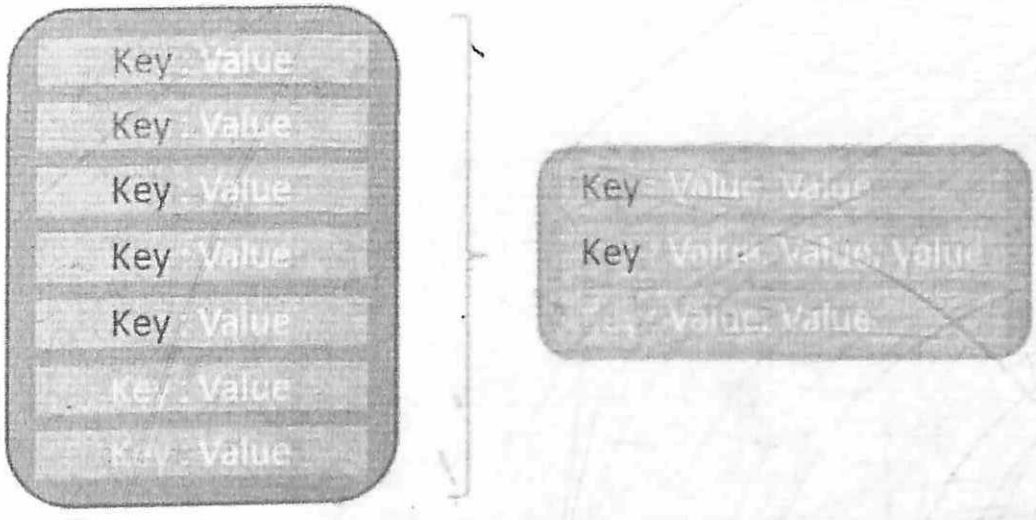
The algorithm consists of several steps. 1. perform a map function for each item in the source collection. 2. Map will return zero or 3.create instances of *Key/Value* objects.

```
ArrayOfKeyValue Map(object itemFromSourceCollection)
```

The responsibility of the Map function is to convert elements of the source collection to zero or more instances of Key/Value objects. See image 3.2.



In the next step, the algorithm will sort all $Key/Value$ pairs and create new instances of objects where all values will be grouped by key. See image 3.3.

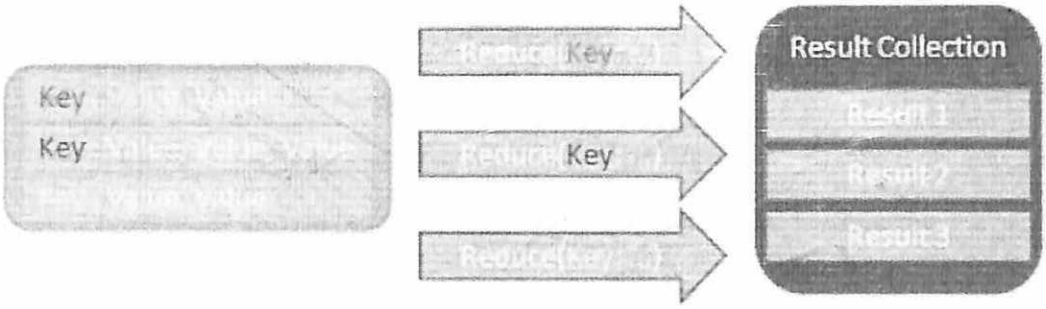


The last step is the *Reduce* function — for each grouped instance of the $Key/Value$ object

```
ItemResult Reduce(KeyWithArrayOfValues item)
```

The Reduce function returns a new instance of the object to be included in

the resulting collection.



Example-implementation

As an example, we implement a very simple and visual implementation of this algorithm in C. My example counts the number of vowels line by line in a rowset.

In the example, we created a generalized function of $MapReduce$, as the main in this algorithm, which simply calls the specialized Map and $Reduce$ functions , threading their implementation. Actually the functions Map and Reduce , the implementation of which is already specific to the problem that we are trying to solve (in each case), and in this case — is to "count the number of vowels in a set of strings."

```
using System;
using System.Collections.Generic;
using System.Collections.Concurrent;
using System.Linq;
using System.Threading.Tasks;

namespace MapReduceSample
{
// Элемент результирующей коллекции, гласная и ее количество
 class VocalCount
{
        public char Vocal;
public int Count;
}

class Program
{
    static void Main(string[] args)
    {
        // "lines" - это исходная коллекция.
        var lines = new[] {
            "How many vocals do",
            "these two lines have?"
        };

        foreach (var line in lines)
        {
            Console.WriteLine(line);
        }
        Console.WriteLine();

        // Вызывается MapReduce
        var results = MapReduce(lines, Map, Reduce);

        // Отображение результата
        foreach (var result in results)
        {
            Console.WriteLine("{0} = {1}", result.Vocal, result.Count);
        }
```

```
    Console.ReadKey();
}

/// <summary>
/// Функция Map считает количество гласных в строке
/// </summary>
/// <param name="sourceItem" />Строка для подсчета</param>
/// <returns>Коллекция экземпляров Key/Value.
/// Где key - гласная, и значение ее количество.</returns>
static IEnumerable<KeyValuePair<char, int>> Map(string sourceItem)
{
    return sourceItem
    .ToLower()
  .Where(c => "aeiou".Contains(c))
.GroupBy(c => c, (c, instances) => new KeyValuePair<char, int>(c, instances.Count()));

}

/// <summary>
/// Функция Reduce считает общее число каждой гласной
/// </summary>
 /// <param name="reduceItem" />Экземпляр Key/Values, где key - гласная,
/// и value - список всех подсчетов гласных по строкам</param>
/// <returns>Экземпляр элемента результирующей коллекции, VocalCount</returns>
static VocalCount Reduce(KeyValuePair<char, IEnumerable<int>> reduceItem)
{
    return new VocalCount
    {
        Vocal = reduceItem.Key,
        Count = reduceItem.Value.Sum()  // Computes total count
    };
}

/// <summary>
/// Обобщенная реализация функции MapReduce
/// </summary>
/// <typeparam name="TSource">Тип элементов исходной коллекции</typeparam>
/// <typeparam name="TKey">Типа ключа Key используемого в функциях Map и Reduce</typeparam>
/// <typeparam name="TValue">Тип Value используемый в функциях Map и Reduce</typeparam>
/// <typeparam name="TResult">Тип элементов результирующей коллекции</typeparam>
```

```
        /// <param name="source" />Исходная коллекция</param>
        /// <param name="map" />Функция Map</param>
        /// <param name="reduce" />Функция Reduce</param>
        static IEnumerable<TResult>  MapReduce<TSource, TKey, TValue, TResult>(
            IEnumerable<TSource> source,
            Func<TSource, IEnumerable<KeyValuePair<TKey, TValue>>> map,
            Func<KeyValuePair<TKey, IEnumerable<TValue>>, TResult> reduce)
        {
            // Коллекция, где сохраним результаты нашей map-функции
            var mapResults = new ConcurrentBag<KeyValuePair<TKey, TValue>>();

            // Выполним функцию Map паралельно для каждого элемента исходной коллекции
            Parallel.ForEach(source, sourceItem =>
            {
                foreach (var mapResult in map(sourceItem))
                {
                    mapResults.Add(mapResult);
                }
            });

            // Сгрупируем все значения по ключам
            var reduceSources = mapResults.GroupBy(
                    item => item.Key,
        (key, values) => new KeyValuePair<TKey, IEnumerable<TValue>>(key, values.Select(i=>i.Value)));

            var resultCollection = new BlockingCollection<TResult>();

            // Старуем reduce
            Task.Factory.StartNew(() =>
            {
             // Выполняем функцию Reduce паралельно для каждого элемента reduceSources
              Parallel.ForEach(reduceSources,
                           (reduceItem) => resultCollection.Add(reduce(reduceItem)));

                resultCollection.CompleteAdding();
            });

            return resultCollection.GetConsumingEnumerable();
        }
    }
}
```

4. Conclusion

In this thesis we consider a modern model of data collection and analysis. Mathematical methods of processing and storage of big data are given. The goals set as a conjugation of the system mathematical model taking into account all possible problem phenomena were recorded and described in full.

Big data practices are required for performance. A number of methods designed to implement data collection and analysis use explicit knowledge representations and carefully designed algorithms to create new products in the market. Today, many large firms and individuals have their own developments in this area.

The following method of working with large amounts of data is considered: MapReduce - development of Google in the field of distributed computing models. Used by the company itself for parallel calculations over large data sets:

References

[1] D Conway R. "Strip Selective partial access to a database". In: *ACM annual conf., New York* (1976), pp. 5–19. DOI: http://dx.doi.org/10.1002/andp.19053221004.

[2] B. Franks. *Taming the Big Data Tidal Wave Finding Opportunities in Huge Data Streams with Advanced Analytics*. 2012, pp. 153–159.

[3] J. Gantz. "The digital universe in 2020: Big Data, Bigger Digital Shadows, and Biggest Growth in the Far East". In: *IDC Country* (2013), pp. 12–18. DOI: https://www.emc-technology.com/collateral/analyst-reports/idc-the-digital-universe-in-2020.pdf.

[4] Chinfy Ozsoyogulug. *Security in partitioned dynamic statistical databases*. 1979, pp. 594–600. DOI: http://dx.doi.org/10.1002/andp.19053221004.

[5] P. Reiss S. "Practical data swapping: the first step". In: *IEEE* (1980), pp. 30–38. DOI: https://ieeexplore.ieee.org/abstract/document/6233701.