Software and hardware infrastructure for timetables scheduling in university

Malikov Andrey
amalikov@ncfu.ru

Vladimir Voronkin
vl.voronkin@raxperi.com

Pevchenko Svetlana
spevchenko@ncfu.ru

Shchegolev Alexey
a.wegolev@gmail.com

North-Caucasus Federal University, Stavropol

Abstract

When making schedule of a large educational institution, we have to work with big data, which requires a special organization of the software and hardware infrastructure: usage of special data structures and indexing, organization of data marts and analytical services, development of new data manipulation algorithms. The paper examines certain aspects of the software and hardware infrastructure for business process of training schedule formation in North Caucasus Federal University.

1 Introduction

Schedule formation is computationally complex problem of distributing finite sets of resources [1], usually non-polynomically complex. The need for scheduling arises in various areas of human activity and technical systems: the traffic schedule, the schedule of work of institutions and employees, the study schedule, multi-threaded multiprocessing, the production line schedule, and so on.

There are two groups of methods and algorithms for automatic schedule formation can: deterministic and stochastic [1]. Examples of deterministic algorithms are dynamic programming [2], optimization of the cost function with preprocessing and without preprocessing [3], estimating the heuristics of multi-threaded tasks in multiprocessors [4], etc. Examples of stochastic algorithms are genetic algorithms [5], various stochastic flow algorithms [1], etc. Automatic scheduling algorithms work effectively in technical systems, for example, when planning the distribution of tasks in multiprocessor systems, but they are of little use when allocating resources in social systems, including study schedules. The schedule formation is characterized by a large number of heuristics and internal agreements existing in any social system. It is also necessary to take into account the availability of "best practices" applied in various educational institutions, and often significant differences in the regulatory framework and approaches to scheduling in different countries. Very often educational institutions, especially high schools, uses software that allows to partially automate this process in such a way that the machine decides the well-formalized computing tasks, and the person implements those processes of resource allocation that the machine cannot perform due to the complexity of the algorithm being implemented and / or the existing conflicting restrictions on the resources allocated.

There are many information systems, adapted for scheduling in educational institutions of the Russian Federation, for example, [6], [7], [8]. Some systems have partially automatic mode or a consulted mode of scheduling,

In: Marco Schaerf, Massimo Mecella, Drozdova Viktoria Igorevna, Kalmykov Igor Anatolievich (eds.): Proceedings of REMS 2018 – Russian Federation & Europe Multidisciplinary Symposium on Computer Science and ICT, Stavropol – Dombay, Russia, 15–20 October 2018, published at http://ceur-ws.org

but as a rule the main way of compiling is a manual method. In this case, known information systems have the following limitations:

- require considerable time to prepare the initial data in the form of the department's study load;
- poorly adapted to the processes of continuous change of the initial data in the study load or do not support the temporality of the initial data;
- do not allow to make a schedule in the presence of individual learning paths for each student, including the limitations when working with large data;
- not convenient system of parallel access of a large number of employees, who making the schedule (next – schedule maker), to shared resources (classrooms, training groups, teachers) using the classical query model of data manipulation.

The software and hardware infrastructure of training schedule formation of the North Caucasus Federal University is considered, which allows solve these problems.

2 Problem Statement

The North Caucasus Federal University (NCFU) is the largest university in the North Caucasus, formed in 2012. Currently, more than 25,000 students study at more than 400 educational programs in the NCFU. To ensure the main business processes of educational, scientific and educational activities, the University uses the integrated automated management system (IASY VUZ). In the database IASY VUZ, since 1994, information about more than 500,000 students have been gathered. The total size of the managed dataexceeds 1TB.

Relational OLTP-database IASY VUZ contains more than 1000 tables and indexes based on B+trees. As a result of the normalization, the metadata schema is represented by a tree:

- the root of the tree is the table *kartapromat* with students, which accumulates data on the features of each lesson and schedule, assessments, attendance, etc.; the table contains more than 2 billion records;
- a subtree with educational programs is represented by tables with the description of educational programs, schedules of the educational process, educational disciplines, lists of classes;
- a subtree with students is represented by tables with personal data of people, training groups, information about students;
- a subtree with employees is represented by tables with the structure of educational institutions, states, information about employees.

The presented scheme supports the individualization of the learning process of each student, associated with the presence of elective disciplines in the curriculum, the availability of several alternative educational programs, the ability of the federal university to create new own educational standards. The curriculums for different education level and form contain from 2000 to 4000 separate lessons with an average of 1000 lessons per year. As a result, it is required every year to schedule for $p = 25000 * 1000 = 25000000$ records. With the modern development of computer technology with such a large amount of initial data, the task of automatically or manually compiling a timetable for an acceptable time seems difficult.

Heuristics are required, which can significantly narrow the area of data to be processed. The main approach is based on the formation of student learning streams in accordance with the constraints imposed on the academic disciplines, activities, classrooms used. The learning flow is a virtual aggregation of individual lessons of students that can be held together when the conditions for the coincidence of the restrictions accepted in the educational institution are met. Examples of such restrictions are:

- the coincidence of the name or content of the discipline;
- complete or partial coincidence of the calendar plans of the educational process;
- restrictions on the maximum number of students in the stream, for example, because of safety requirements;
- restrictions on the maximum number of students in the stream due to limitations of the classrooms.

3 Business-Process of Schedule Formation and Creating Study Streams

The business process of preparing the study schedule can be conditionally divided into the following stages:

1. The formation of an educational program;
2. Formation of a contingent of students on the educational program;
3. The distribution of students in elective disciplines;
4. Calculation of training streams;
5. The distribution of compulsory resources for educational streams: teachers;
6. Updating the data mart with the study load;
7. Distribution of shared resources by educational flows: classrooms, time;
8. Mapping of the timetable to individual learning trajectories and the formation of individual schedules for students and teachers.

Figure 1 shows the main components of the database that provide the business process for scheduling.

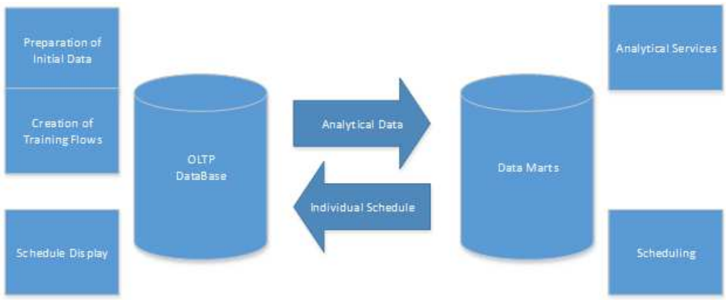

Figure 1: The main components of the database that support the business process of scheduling

There are two models for calculating learning streams:

1. In the initial calculation mode, current regulations and restrictions are taken into account when forming the stream. The main tasks at this stage: to minimize the number of streams within the limits of current restrictions; as much as possible to reduce the number of student permutations between different streams to avoid unnecessary checks of individual overlays in the schedule for each student.
2. In the regime of continuous updating of streams, changes in educational programs and movement of students are taken into account. The main tasks at this stage: if possible, to preserve the initial set of streams when students are added and removed during the year, including possibly with violation of certain restrictions; when distributing new students through the streams, first of all, only information about the stream should be taken into account, but not the norms of its formation. At this stage, the study streams have already been checked by specialists and accepted for processing.

The flowchart for calculating the study streams is shown in Figure 2. The algorithm for calculating the study streams is a modified version of the greedy algorithm, which makes it possible to significantly reduce the range of enumeration of all possible combinations of student pooling into streams, but on the other hand does not guarantee global optimization of the formation of streams. At each step of the algorithm, the problem of local optimization is solved by comparing each new occupation of each student with the already formed streams, sequentially checking from the smallest streams to the largest ones. If a suitable flow is found, the student is added to it, otherwise a new stream is created from one person.

The algorithm can be effectively paralleled when partitioning the sorted list in step 2. At the same time, sectioning should be carried out in such a way as to ensure that all the activities of one student and all potential for the organization of students stream into one section fall into place. Examples of such sections may be institutes or faculties in situations where interfaculty streams are not used in an educational institution. When using the software and hardware infrastructure of the NCFU, the parallel implementation of the presented algorithm takes 8 hours if 15-20 processes are organized.

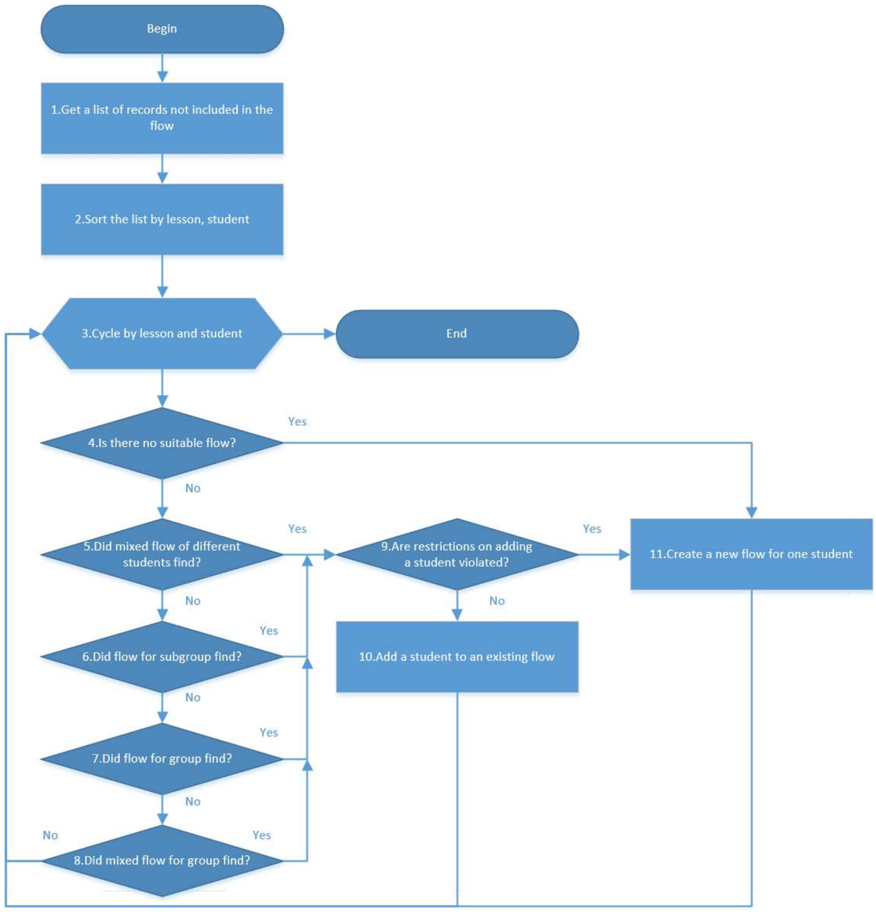

Figure 2: Flowchart for calculating study streams

4 Creating a Data Mart

In order to perform analytical services and organize automated workplaces for schedule makers, a data mart with study streams organized in the database. The main implemented analytical services are:

- calculation of the load of basic resources: classrooms, teachers, training groups;
- calculation of hourly pay for teachers;
- optimization and load balancing of students and teachers;
- calculation of the efficiency of the use of scientific equipment and laboratories;
- optimization of the work of auxiliary services: food, security;
- optimization of the use of water, electricity, etc.

The data mart contains pre-processed aggregated data on study streams and their properties and is designed to update the multidimensional cubes of the OLAP-module of the information system built according to the MOLAP scheme with the update period – one time per day. The main measure is the number of students in the subgroups, combined into a stream. The measurements are: educational programs, groups of students and study streams.

The structure of the fragment of the data mart with a table of measures and tables for storing information about the study streams and groups of students is shown in Figure 3.

The data mart does not store information about individual trajectories of students learning. In order for the information system to track the overlays for each student, the data mart is supplemented with a table

$'matchFlows'$ (kod_pot1, kod_pot2) to store the symmetric binary relation of the prohibited overlays in the schedule. Each entry in the table contains all pairs of a binary relation, while only $kod_pot1 < kod_pot2$ pairs are stored with symmetry taken into account. To fill the table $'matchFlows'$, you need to match all the pairs of lessons of students to each other, i.e. perform $C_p^2 = C_{25000000}^2$ comparisons. The computational complexity of pairwise comparisons is described by a polynomial of the second degree. To reduce the computational complexity, the list of activities can be compared to itself using a merge join, i.e. doublepass the sorted list. In this case, the computational complexity corresponds to twice the length of the list, but you will also need to sort the list and hash the result to obtain unique pairs of threads with shared students. Script in SQL to fill the table $'matchFlows'$:

```
--the temporary table #temp maps to each flow kod_pot a list of its students
kod_cont
--the table dbo.kartapromat contains a list of all activities for all students
select distinct sl1.kod_pot, k1.kod_cont into #temp
from sch.scheduleLog sl1 inner join dbo.kartapromat k1
        on k1.kod_ng=sl1.idSchedule-- time-sensitive records
where getdate() between sl1.dateBegin and sl1.dateEnd and
k1.pr=1-- a sign of elective courses for students
-- build B + tree
CREATE INDEX IX_kod_cont ON #temp (kod_cont, kod_pot) ON [PRIMARY]
-- determine the flows in which there is at least one common student
insert matchFlows (kod_pot1, kod_pot2)
select distinct t1.kod_pot, t2.kod_pot
from #temp t1 (nolock) inner join #temp t2 (nolock)
        on t1.kod_cont=t2.kod_cont and t1.kod_pot<t2.kod_pot
drop table #temp
```

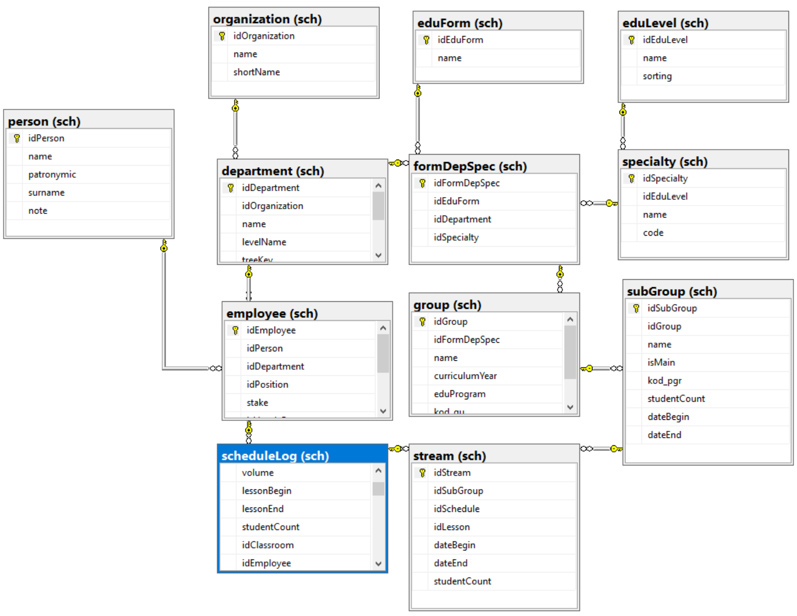

Figure 3: Fragment of the data showcase with learning flows

Figure 4 shows the plan of the last query in the script. When performing a merge join, the connection can be parallelized into several processes by cutting each of the two sorted copies of the student list.

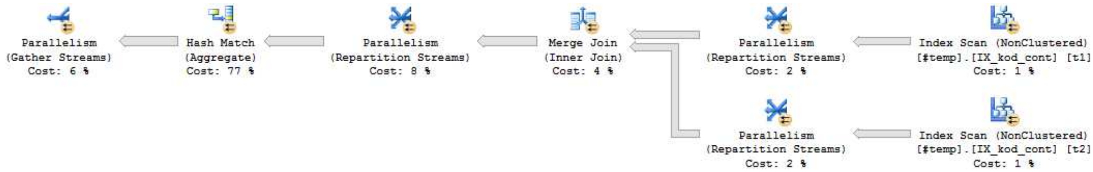

Figure 4: Plan for calculating overlays in the schedule for each student

5 Software for Timetables Scheduling

The architecture of the program module used by the staff of the university for scheduling classes consists of three main components, represented in Figure 5.

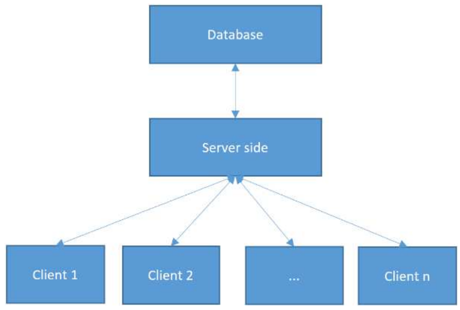

Figure 5: Software module architecture

1. Database

   Database, as well as related information processes were described earlier.

2. The server part

   The server part of the software module (Services) is the intermediary between the database and client applications. Its tasks are: transparent access of clients to the database, ensuring the simultaneous operation of all schedule makers and synchronization and data exchange between clients.

3. The client part

   Client part allows: calculate conflicts, arrange classes and view the load of classrooms, teachers, departments.

The architecture of the software module is designed in such a way that all schedule makers work in a single information space. Changes made by one schedule maker are instantly visible to all other schedule makers. To avoid creating conflict situations in the schedule grid (for example, when the teacher at the same time is engaged in several groups), the software module has a subsystem for calculating the conflicts. With any changes to the schedule grid, the conflicts are automatically recalculated for all clients.

According to this approach, the dispatchers that make up the schedule can track changes in real time and see any kinds of overlays.

The client part consists of the following subsystems:

1. Subsystem for calculate conflicts

   The subsystem for calculating the conflicts is a module designed to account for conflicts in the schedule. The following types of conflicts are possible:

   (a) the conflict by the teacher: the teacher conducts the lesson at the same time for several groups;

   (b) the conflict by classroom: in the classroom in one lesson, several groups are put;

   (c) the conflict by the number of seats in the classroom: the number of students exceeds the number of seats in the classroom.

2. Storage subsystem

   Calculation of conflicts requires storing all data associated with the open schedule grid on the client and updating conflicts, when changing data, by other schedule makers.

3. Updates subsystem

   The database contains all records of student movements (transfer to the course, enrollment). In order for schedule makers to have flexibility in scheduling, they can selectively make some changes made in database. This process is called acceptance of updates.

   Some updates can make changes to the already arranged schedule. In order not to make changes to the schedule already created, they can selectively receive updates.

The procedure of accepting updates is changing the field that stores the date when updates were last received. Figure 6 shows a section of the database schema.

The update is considered accepted if the $lastUpdate$ entry is less than $versionDate$.

$VersionDate$ is the version in which the schedule is running. To accept the update, you must change the $lastUpdate$ entry to the current date, which is always greater than $versionDate$.

After accepting updates, all changed records are automatically transferred to all clients.

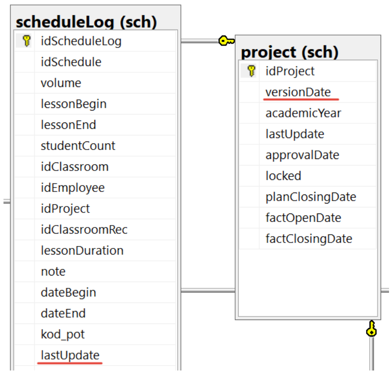

Figure 6: Part of the update acceptance database

6 Discussion Results

The developed software module is a unique solution at the moment, which allows to make a schedule, take into account overlays, view the load of classrooms, teachers and departments; form various reports; view the schedule through the Internet.

The schedule is available on the Educational portal NCFU http://eCampus.ncfu.ru in the section "Schedule" [9]. Unauthorized users have access to a schedule of streaming lessons, including the schedule of groups of students, teachers, classrooms (Figure 7). After authorization, the user is granted access to a personalized individual schedule.

The client part of the software module allows to create a schedule using the usual gestures drag'n'drop. To add a lesson to the schedule grid, schedule maker needs to drag the lesson from the left side of the application to the schedule grid. To add a classroom to the lesson, schedule maker needs to drag the lesson from the right part to the lesson on the schedule grid. Some classes are assigned to the classroom, and when dragging these classes to the grid, the classroom will automatically be added to them.

The appearance of the program module is shown in Figure 8.

7 Conclusion

The paper presents an approach and describes the software and hardware infrastructure of the big data processing NCFU in the task of calculating the study load and scheduling lessons. Taking into account individual trajectories of training, it is necessary to process several tens of millions of records per year. To solve this problem, special algorithms have been developed for narrowing the area of processed data, forming study streams, calculating

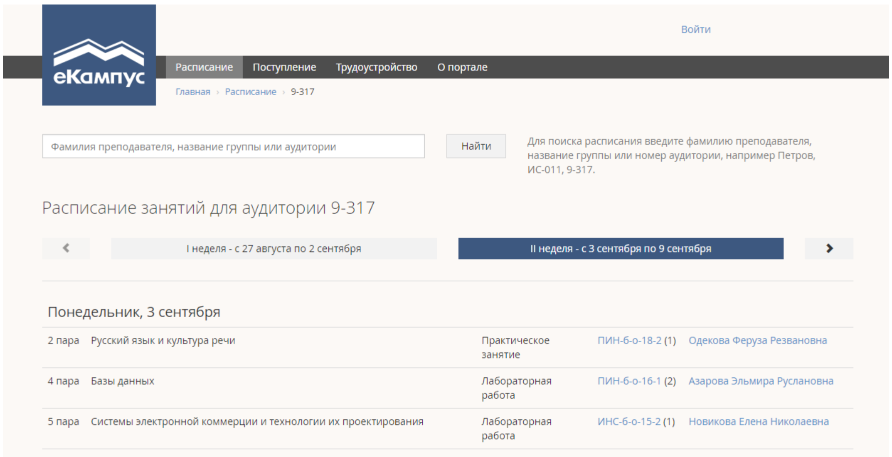

Figure 7: Example of displaying the schedule of classes on the Education portal of NCFU

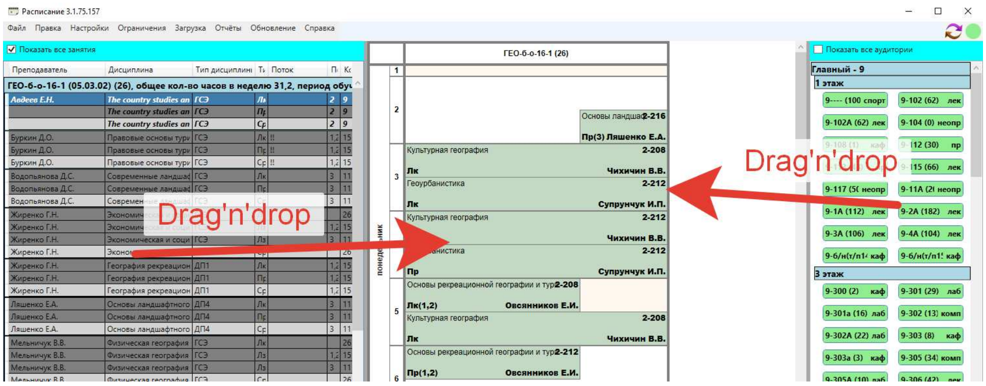

Figure 8: Appearance of the Information module

overlays in the schedule. To work with a limited set of study stream records and provide analytical services, a data mart has been developed and updated.

At the final stage, the opposite task is solved: the timetable is mapped on individual trajectory of training, which allows to form an individual schedule for each student and teacher.

References

[1] Pinedo Michael L. *Scheduling: theory, algorithms, and systems.* Springer Science+Business Media, LLC 2016.

[2] Bomberger Earl E. *A Dynamic Programming Approach to a Lot Size Scheduling Problem.* Management science, volume 12, Issue 11, 1966

[3] Hadas Shachnai, Tami Tami, Gerhard J. Woeginger *Minimizing Makespan and Preemption Costs on a System of Uniform Machines.* European Symposium on Algorithms ESA 2002: Algorithms — ESA 2002, pp. 859-871

[4] Santos D.L., Hunsucker J.L., Deal D.E. *An evaluation of sequencing heuristics in flow shops with multiple processors.* Computers & Industrial Engineering, Volume 30, Issue 4, 1996, pp. 681-691

[5] Vincent Van Peteghem, Mario Vanhoucke *A genetic algorithm for the preemptive and non-preemptive multi-mode resource-constrained project scheduling problem.* European Journal of Operational Research, Volume 201, Issue 2, 2010, pp. 409-418

[6] *Automated information system for scheduling classes in educational institutions.* $http : //www.tolgas.ru/org_structura/kaf\_iis/nauka/project/raspisanie/$

[7] *BIT.VUZ. Schedule*. http://www.pulsar.ru/progs/1904/

[8] *Express schedule School Mini.* $http://pbprog.ru/products/programs.php?ELEMENT_ID = 365$

[9] *Educational portal of NCFU.* $http://eCampus.ncfu.ru$