International Journal of Engineering Education Vol. 39, No. 5, pp. 1207–1215, 2023
Printed in Great Britain
0949-149X/91 $3.00+0.00

Development and Assessment of a New Approach to Teaching Parallel Databases*

WEIHONG ZHAO[1], MERUERT SERIK[2] and SAULE ZHUMAGULOVA[2]**
[1] International Business School, Huanghai University, Qingdao, China. E-mail: weihongzhao53@gmail.com
[2] Department of Computer Science, L. N. Gumilyov Eurasian National University, Nur-Sultan, Kazakhstan.
E-mail: meruert_serik@rambler.ru, saulezhumagulova52@rambler.ru

Software development, database management and parallel computing are deeply integrated and related technologies. Both types of technology are rapidly evolving and becoming more sophisticated, with resulting subtleties of their interaction learned directly in the workplace. Academic research focusing on developing programs or approaches to professional training in these areas in universities is relatively scarce. This study provides a theoretical background and practical approach to building parallel database programming skills by introducing software developers' best practices. The piloting of the proposed approach was attended by 64 third-year IT students in the experimental group, who were involved in 3-month training, with their learning pathway depending on the challenges they faced. Tasks were assigned and checked by instructors. The findings suggested that this approach can help to improve technology proficiency among students in accordance with the recognized professional competence level rather quickly. No one achieved such a result in the control group. The Student's t-test showed that the differences in scores between the control and experimental groups were statistically significant. Essentially, the research findings can be used to teach parallel databases and to strengthen ties with graduates' potential employers in the corresponding academic programs.

Keywords: parallel computing; programming; databases; programming language; student education; parallel computing education system

1. Introduction

Managing relational databases during network interactions is becoming one of the most important and demanding programming skills for real-world business tasks [1, 2]. As many of the business programming tasks in marketing data processing, user relationship management and other applications involve the use of vast arrays, the challenges facing database designers and managers are changing. Artificial intelligence involves the use of deep learning, design of neural networks and the use of heuristic algorithms. Deep learning processes most often cannot be controlled by humans, providing results in the form of a black box. This enables testing the adequacy, accuracy and applicability of the deliverables, but does not provide a way to assess how such deliverables were ultimately obtained [3]. With this in mind, most deep learning processes are built on data-driven approaches, which are increasingly recognized as dominant in problem-solving techniques [4, 5].

Data-driven approaches require increasingly large arrays, Big Data technology and, consequently, large databases. In commercial contexts, a significant portion of such databases are distributed and perform much of the query processing and data storage on the network, using cloud computing and distributed computing [6]. When it comes to training in database programming and administration, important issues emerge. Applications programmers typically have trouble navigating the specific problems of optimizing the processing of large arrays. Experts in database design and application who would like to test their approaches on large samples of real-world data or with valid user apps also face challenges [3]. The solution to this problem is seen in special high-level query languages that make possible an abstraction of most low-level data and ideas, which should not distract from the database using tasks. The problem comes from the fact that meeting these needs requires skills and knowledge that far exceed the typical skills of both of the described professional teams.

Such special languages were recommended for study starting at the most basic level of programming skills in order to train a good professional in a university [6, 7]. This also opens the way for the immediate involvement of future programmers in learning innovative paradigms to support embedded parallel databases implied by new query languages [1, 8]. The new paradigm of database learning and presentation involves a number of fundamental declared ideas: focusing on first-ordered query language; application of database management systems (DBMS); focusing on the deductive database paradigm (DDB).

Parallel computing is a common approach to

** Corresponding author: saulezhumagulova52@rambler.ru

* Accepted 1 June 2023.

dealing with major challenges of implementing innovative database design and operation paradigms [9, 10]. Their commercial, rather than emulating, implementation involves the use of multicore processors that support simultaneous running multiple threads and atoms in computing. This way, the following issues are addressed: parallel query processing; ranking the level of access to different areas of the database; making sure that cryptographic protocols function properly; speeding up access to data from different physical addresses; distributing the load on the various database servers, etc. [9].

The practical implications of training students in database management involve a group of similarly complex problems described in a number of studies [10–12]. More specifically, the training and the use of software technologies occur on relatively small samples of data and a small number of queries, which should emulate the database activity. In real-world contexts, professionals have to deal with very large volumes of data and very large numbers of queries. Coping with these problems is crucial to database activity [11]. Configuration set-up is vital for optimizing the database management system's performance. The configuration task becomes complex and non-trivial for Cloud Databases (CDB) because it involves different database instances, different workloads from queries [6, 13]. Various types of CDB configuration automation have a number of system limitations, which are also best introduced to future professionals well in advance [5].

Contemporary studies on training to use parallel computing in the design, optimization, configuration and administration of databases in the educational aspect are considered rarely. Rapid development of programming technology and the server hardware, as well as increased computer performance resulted in a situation where in most cases training is conducted directly in the workplace while solving real-world problems [7, 14].

Closely related to this problem is the problem of Big Data mining, addressing which requires a streaming configuration of the database in accordance with the requirements and structure of the accumulated information [15]. Whereas previously mining was based on the pre-configuration of databases for a particular set of data and their intended areas of research, now distributed databases are increasingly used to analyze data of various formats and content. Such data can be analyzed in different ways, according to different algorithms, and for various needs [4, 15]. This feature requires database administrators and programmers to be skilled in managing processes while simultaneously optimizing and configuring databases, as well as formatting parallel queries using special query languages [8].

Yet, it seems possible to identify a long range of technologies, as well as already developed and validated approaches to parallel computing in the case of distributed databases, which can be used as a basis for early practical training and providing theoretical knowledge to IT professionals. Such training will significantly reduce the time required for the future professionals to start solving commercial tasks and will increase the efficiency of IT companies.

1.1 Literature Overview

Available research papers address the problems of configuring distributed databases with parallel computing by employing many special methods. Most of them discuss data management options, brought about by the fact that the state and performance of specific computing devices in distributed and cloud databases are unpredictable, and the load on database nodes is also stochastic [1, 11]. This problem requires consideration of not only software, but also engineering and technological aspects of the problem in their connection with software engineering.

Research in engineering education points to the need to develop real-world problem-solving skills in engineering students and, in this regard, research on problem-based learning is expanding [16, 17]. The solution of real problems demonstrates the complexity of engineering calculations and creates the need for both the distribution of tasks and teamwork, as well as the need for psychological support for team members, as well as the integration of Data Science into the curriculum of technical and engineering specialties [18]. Data management and management of large data sets are currently a critical part of the training of engineering specialists not only in the field of software engineering, but also in other areas. Therefore, researchers emphasize the importance of interdisciplinary learning and solving engineering problems using software methods and acquiring engineering knowledge and skills by programmers [19, 20].

Data management requires computing devices that can process data to generate better information. Parallel computing systems can be grouped according to the level at which the hardware supports parallelism. Yet, contemporary studies offer a significant number of classifications of parallel computing, which include various forms of ensuring real parallel computing at the device level. These are Massively Parallel Computing, Graph Processing Unit (GPU), Distributed Parallel, Multi-Core Processors [1]. As a rule, the solution of simultaneously engineering problems at the level of hardware and infrastructure is rarely considered in close connection with the corresponding software problems [19, 21].

Massively Parallel Computing is the most well-studied technology within database methods because it emerged and was implemented for single-core processors, when real parallel computing was achieved using a large number of networked computers [22]. Such an organization requires solving practical engineering problems, ranging from the design of excess heat removal from large groups of processor units and memory devices to the architecture of the room and ensuring the optimal power supply scheme. This situation is now commonplace, with numerous technologies for its implementation that automate most of the regulatory processes and the configuration of computing processes. However, the situation known from MPC-related expertise is complicated by the fact that it serves as the first level of a sophisticated structure that includes the following forms of parallel computing [23].

Traditional empirical database optimization techniques (such as index and view selection, cost estimation, knob tuning, join order selection, and others) do not meet the requirements pertaining to increased performance in the case of large-scale database instances involving diverse applications and which include diversified users, especially in cloud database environments [10]. Performance of database algorithms may be improved by increasing the accuracy of the frequent item set as one of the options of association rule algorithms [15] stands out. The problem may be solved through mutual optimization of control algorithms of applications and the structure of databases through a number of complex structured processes.

Teaching methods and approaches pertaining to databases and parallel computing are an important component of the studied problems. Neo-Piagetian approaches to learning, which combine constructivist methodology with contemporary teaching approaches, are most common in academic studies. A comparative analysis of objectivist and constructivist approaches by a number of researchers suggested that constructivist methods can seem intimidating and complicated for many learners, and in some cases didactic teaching proves more acceptable [12, 24]. This is especially true of extreme constructivist approaches in which learners have almost unlimited freedom in structuring their own learning process. The Neo-Piagetian approach takes into account the requirements imposed by the real-world contexts and suggests structuring the learning process based on such requirements, but taking into account the personal choices, pace, and learners' traits [25]. The implementation of this principle depends on the type of learning and the complexity of the learning environment and the degree of its virtualization [26]. If students are highly motivated and involved in the learning process, a fairly high task personalization degree is possible. Such personalization is achieved by performing tasks with different types of parallel queries and solving a large number of typical problems that arise from the way the virtual server handles streams of parallel queries with different workloads. The variety of such types of assignments contributes to meeting a very wide range of requirements for varying levels of difficulty, the topic being studied, and the learners' success [27-29]. Inclusion of virtual server and cloud computing assignments to develop tasks and training programs for IT students and database administrators seems necessary, based on the existing requirements of applied programming. It is essential to simulate the situation of a large number of queries with a large amount of data and create virtual database instances with a large number of entries to allow students to solve the real-world programming task. Implementing such an approach requires a large number of typical tasks for implementation of close-to-reality parallel queries to databases. Accordingly, it is also essential to have characteristic features of databases, as well as their configuration and content to be emulated during class assignments [6, 30].

Many countries encounter mismatch of important industrial databases amid increased complexity and the growing number of parallel queries [5]. Shenzhen Emergency Management Bureau has encountered significant differences between various network participants and database servers of industrial and administrative facilities. The hardware of many servers is obsolete. Different interacting databases rely on standards of different versions. There are differences in the system architecture of information systems. Data exchange, attempting to implement parallel computing, as well as sharing internal and external data did not rely on a common standard. A common platform for data and centralized data management was not set up as well [5]. As suggested by available research, the described situation is standard rather than exceptional for many countries [6, 30, 31]. Such developments bring to the forefront the issues of training IT professionals capable of re-engineering databases for their interaction in cloud services and for implementing massively parallel processing at a very sophisticated level. This means that they must become aware of such complex issues in the early stages of their education and learn to solve them as they develop knowledge and skills. The constant change in the hardware and architecture of specific databases created to solve specific problems is a challenge for the training of engineering specialists and the formation of the necessary engineering knowledge among programmers. Programmers must have engineering knowledge in the required quantity to ensure the flexibility and changeability of the architecture of design solutions [22]. Therefore, being aware of the depth and complexity of real-world problems should shape and drive the personalized curriculum structuring by each student. Personalized programs can then come not from subjective and often inadequate perceptions of the professional needs and challenges, but from a strong awareness of real situations that require resolution [21].

The novelty and contribution of research in the field of engineering lies in the proposal of a method for involving engineering knowledge in the training of programmers to accelerate the adaptation of new engineering solutions to the construction and operation of modern databases. Because of existing gaps in research of sophisticated interrelated technical skills, such as parallel computing and database management and hardware engineering, this paper proposes a methodology for providing a theoretical background and practical training of future professionals under various academic programs.

2. Method and Materials

2.1 Participants

For the purposes of the study, a group of 64 third-year IT students was set up based on random sampling of IT students. All group members wanted to become proficient in parallel databases and were willing to be involved in testing the experimental learning approach. The group consisted of 37 men and 27 women, aged 21–28 years.

2.2 Research Design

Over the course of 3 months, the control group received the course of parallel computing in databases, which is usual for this educational institution, without any changes. The experimental group was trained according to the same program, without the exclusion of any elements, but using a new approach to learning. The experimental group members were offered a new learning approach, which did not provide for significant changes in the existing class schedule or the content of the courses already in place. The differences were as follows:

The accuracy and completeness of the provided information were discussed with the companies' programmers and engineers in order to avoid the disclosure of trade secrets, while providing students with the most complete information about the problem and its complexity.

Three online discussion sessions were held on the Delphi Method. Following such discussion, 15 tasks approved by all contributors to the discussion (in terms of complexity, innovative technology and the companies' needs) were selected. The task complexity was chosen so that a competent programmer would require 15–20 hours to solve each of them. Discussion contributors checked the tasks' compliance with these requirements relying on internal statistical data of the companies' management.

The companies' IT professionals involved in design of the training tasks used heuristic algorithms to generate real-world databases and tool-related requirements that corresponded, as closely as possible, to the real-world contexts faced by the companies participating in the experiment. The group of IT professionals included system administrators, programmers and engineers involved in the creation and configuration of parallel databases, from engineering the server hardware and server architecture to high-level database administration. The tasks they proposed were aimed at ensuring that students were ready to solve problems associated not only with the formation of queries or database security, but also problems generated by design features and engineering solutions. To do this, the experimental group was offered the necessary engineering knowledge related to the architecture of parallel databases. That way, the training setting was as close to the real-world context as possible. The problem's input data were not emulated in a simplified and abbreviated form, but actually corresponded to the actual data. The pre-agreed capacity of university servers was used to perform the tasks. Therefore, students worked with a real database of 200,000 entries with several hundred incoming queries simultaneously, which had to be processed by 2 local servers (one from each of the universities whose students were involved in the experiment), and virtual server based on Azure technology, on which the distributed database was stored. Each server had its own performance and architectural features. Possessing considerable hands-on experience in the principles and skills of implementing parallel computing made it possible to quickly recreate (as close to reality as possible) such constructs for each task. Due to the limited computer performance, students had to solve problems step-by-step rather than simultaneously, which, of course, is a certain limitation.

These tasks were presented to the experimental group members at the beginning of their experimental course as a challenge. The participants were given full information on the tasks, as well as on all necessary and referenced technologies, tools, apps and databases to solve the problem. Yet, they were not informed about how each task was solved.

Further on, the participants had to significantly modify their personal learning pathway, focusing on the requirements that the task set for them. The timing was sufficient, though not excessive. All tasks required awareness of additional technologies and tools, which participants could choose and learn independently. The amount of knowledge that they were receiving within their university's regular program, taking into account the additional special knowledge that they had to learn independently, was sufficient to solve the problems. However, the overall cognitive load did not increase, because study participants could choose courses that matched their focus and reject those that did not meet their needs at the time.

In order to compare the effectiveness of the experimental teaching approach, an additional research was conducted in a group of 64 third-year students who did not participate in this study, but who learned parallel databases independently, as part of regular university courses. Members of this group were selected after an interview that made sure they had the knowledge and skills necessary to successfully approach the tasks at hand. They were given to solve the same 15 problems on parallel databases, designed by experienced software specialists. The requirements regarding performing the task in this group, which can be considered the control group, corresponded exactly to those of the experimental group.

2.3 Data Analysis

Statistics were collected on the results of problem solving in the control and experimental groups. Solutions were tested within the context of each specific task. A task was considered to be solved if the software suggested by the trainee performed the functions required to solve the problem (the speed of data transfer, the number of simultaneously processed queries, elimination of specific difficulties in the database, etc.). The quality of solving each specific task was evaluated on a 5-point Likert scale. Within a given scale, the scores were as follows:

- 1 point – ineffective solution that generally meets the boundary values;
- 2 points – a more effective solution that requires significant optimization;
- 3 points – a minimally sufficient solution for commercial application, with sufficient average efficiency and which can be optimized;
- 4 points – good optimized solution which may be improved further;
- 5 points – excellent solution; the company will be ready to hire the solution's author immediately.

2.4 Ethical Issues

No personal information about study participants was collected, analyzed or stored during or after the study. Confidentiality of information on involvement and achievements of all participants was guaranteed. The results of the experiment had no effect on the students' current grades and exam scores, because they concerned specific areas of their majors. Data on participants' achievements were identified using special unique identification numbers that were randomly generated for each participant at the beginning of the study.

2.5 Research Limitation

The study was limited by the available IT equipment that could be used to provide, as fully as possible, the learning task context for the participants. This determined a number of limitations. More specifically, participants could not choose the task sequence or the time they wanted to spend on each specific task. That way, the personalized learning track was somewhat tied up. Furthermore, participants were not provided with extensive special methodological or pedagogical assistance in connection with the special assignments they received, although they could seek advice from their professors.

The study did not focus on demographic (gender, social, age, etc.) differences among participants as a possible factor in their effectiveness, as this might have been a goal of another study.

The scores earned by the members in both groups were compared by both descriptive statistics, i.e., by the number of completed tasks, and by the average grades received for the completed tasks. In order to assess whether the differences between the members of the control and experimental groups are statistically significant, the Student's t-test was used. That way, the study's pre-formulated ***null hypothesis*** was tested. There is no statistically significant difference between the average expert evaluations of the experimental and control group members on each of the offered tasks on parallel databases.

In order to compare the difficulties in solving each task, direct correlations between the scores for each task in the two groups were addressed. The Pearson correlation coefficient was used.

3. Results

Both groups coped with the tasks with great difficulty. Not a single task was completed by more than a third of the participants (at most, the 13th task was completed by 52 participants out of 64 in the experimental group) (Table 1). The average level of completion of any task is dramatically lower. This

Table 1. Comparison of the number of students who completed the task in the control and experimental groups (Pearson correlation)

Task number	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Exper.	32	28	16	34	17	8	9	11	27	18	29	14	35	24	21
Control	16	18	19	22	14	11	7	12	21	9	8	7	11	12	7
r	**0.442**														

Table 2. Comparison of the average scores given by experts for each task in the control and experimental groups (Pearson correlation)

Task number	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Exper.	3.16	2.56	2.61	1.89	1.75	1.78	2.19	2.13	1.03	3.06	3.19	2.49	2.19	1.44	1.16
Control	1.13	1.16	1.09	1.01	1	1.11	1.31	1.51	1.14	1.79	1.56	1.38	1.14	1.15	1
r	**0.577**														

further suggests that there is a significant demand for intensifying the training of IT professionals in dealing with the real-world challenges faced by software companies. Parallel computing and its use in interaction with cloud services and databases is developing very rapidly and continuously requires additional training and proficiency in new practices, tools and methods. During the experiment, the students were not put in an exceptional situation where they were required to possess knowledge and skills that they did not already possess. Instead, they found themselves in a situation typical for IT companies, when parallel databases-related tasks faced by programmers require significant additional efforts to learn additional data and tools. Most participants were not prepared for this situation.

Table 1 suggests that, on average, more students in the experimental group coped with most of the tasks. There are only three tasks with which more students in the control group coped as compared to the experimental group. The coefficient of correlation between the number of students who coped with the tasks is estimated as average (r = 0.442). Absence of significant correlation between the number of students who coped with the same task in the experimental and control groups might suggest that better performance of the experimental group does not result from the content of the tasks. In this case, the correlation would be dramatically stronger. Student's t-test validated the assumption that the proposed approach to learning caused higher success of the experimental group. Some correlation can certainly be observed, but in this case it should not be taken for granted. There is no reason to conclude here that the same tasks caused the same difficulties for both groups of students, Therefore, the context and content of specific tasks corresponded to the overall level of training received by both groups and made it possible to solve the tasks with other supporting factors.

Table 2 clearly demonstrates the differences between the groups in terms of the quality of completed tasks. On average, the quality of tasks in the experimental group is dramatically higher. Yet, it remains very poor from the perspective of the expert group that made the assessment. Only in one case (task 10 in the experimental group) the average score amounts to 3 points, ensuring compliance with commercial quality standards, in which case the task will not necessarily be finalized and optimized before delivery to the customer. This aspect of the study suggests that despite the good results demonstrated by the experimental approach in the experimental group, the group's level on average is not sufficient for professional solution of real-world problems in the studied area. However, an important positive result is that, compared to the control group, which was not challenged by the real-world tasks but had the same general knowledge and skills, some learners in the experimental group achieved sufficient professionalism to solve the task in a surprisingly short period of time.

Student's t-test demonstrates that the differences in the groups' mean scores on the different tasks are statistically significant (p = 0.012), which makes it necessary to reject ***the null hypothesis*** (Table 3).

4. Discussion

Elements of engineering education are critically needed in those areas that are in contact with engineering problems and are designed to ensure the effectiveness of engineering solutions in the real world [22, 32]. First, this is software engineering, which ensures the maintenance of infrastructure, management and control of mechanisms, control of the security of objects, etc. [33]. Also, the programming of parallel databases should consider the features of the hardware architecture and the features of the engineering support of the physical

Table 3. Results of Student's t-test for average scores given by experts

Mean	*df*	*t-test*	*p*
2.32	95	0.677	0.01

structure of the databases to ensure their efficient and flexible functioning [16]. The results of the study demonstrate that incorporating complex engineering knowledge into the process of training software engineers in the form of finding solutions to real-world problems leads to a higher result for training in their professional field. On the other hand, software engineers trained in this way will demonstrate a higher level of readiness to implement new engineering solutions.

Problems with effective training of IT professionals for the challenges they will face in the workplace and in practical engineering solutions are typically addressed in research from the broader perspective of training programs' improved effectiveness, enhanced collaboration and improved individual competencies or thinking skills [2, 11, 34]. Learning how to handle databases or parallel computing is viewed as part of a programmer's regular curriculum. The need to consider the peculiarities of the engineering solution of the database architecture forms the skills of the programmers and future engineers to handle engineering problems in general. The use of virtual reality-based learning examples is also useful for training in engineering education [17]. Methods to provide special skills within these disciplines are assessed [9].

By addressing parallel computing and distributed database technologies, researchers repeatedly pointed to a gap between the needs of these rapidly evolving fields and the skills typical of most programmers and engineers already working in a particular field [22, 24]. Quite often new skills need to be acquired, or technologies for the abstraction of unfamiliar information by an IT-specialist need to be developed. For example, becoming proficient in a new version of an already familiar SQL query language for databases is much easier than becoming an expert in Data Science and machine learning [35]. With the growth of professional knowledge, the test in the development of engineering knowledge related to the provision of software engineering for practical engineering solutions in various fields increases [16, 17].

Development of specific skills through training programs involving machine learning at some level has already been considered. Various forms of AI were used to develop the curriculum, generate learning tasks, and most often, to explore a particular learner's pathway and build an optimal learning environment and program for such person [3]. So far, however, these approaches predominantly optimized specific curricula, developed by instructors relying on own experience or using AI for solving specific problems, as in the case of the proposed academic program [36, 37].

Enhanced learning based on real-world problems and challenges faced by commercial software vendors and database integrators has been used before. However, unlike the proposed methodology, real-world problems were presented as already solved cases to be studied and memorized, or as high-stakes assessment papers [7, 14, 21]. However, presenting sophisticated problems and tasks as a challenge that spurs learning and guides the learner's personalized pathway can be significantly more effective in training [14]. The track record of gradually solving real-world problems is important even more because, as other studies on innovative training techniques have suggested, this approach enhances the ability to solve previously unknown problems [12]. A more precise definition of such learning enhancement opportunities requires a specific study of the experiences of IT program trainees after they got a job [27].

Parallel computing faces a wide variety of challenges. These challenges include geographic information systems (GIS), quantum computing, bibliometrics, distributed systems for large-scale scientific calculations involving multiple computers, securing information in social media or payment systems, managing complex coordinated and high-speed processes (spacecraft, aircraft, large transport and production hubs, and high tech industries) [23, 30, 36]. Most studies examined programming methods rather than teaching approaches to learning them, as it is generally accepted that professionals learn new technology they need in the workplace [14, 29].

Yet, special techniques for learning parallel computing, as one of the basic cutting-edge programming technologies and in connection with real-world tasks, are considered rarely [13, 28, 38]. This is because real-world parallel computing tasks almost always involve processing either an intensive data stream (e.g., database queries), or processing large amounts of information, or optimizing calculations for which the timing of results is critical [9].

Short-term and long-term effectiveness of innovative approaches and platforms (such as specific high-level query languages, tools for decomposition of parallel programs and overhaul of their execution, as well as other methods) in the training of IT professionals has not yet been investigated. Ensuring long-term effectiveness is closely related to changing the engineering decisions of the architecture of data centers and the physical placement of databases, which should be considered by software engineers [17, 32]. This is partly due to the methodological complexity of such studies, recognized by the researchers themselves [8, 39]. Therefore, a special role in this paper is attributed to finding solutions as close as possible to the real-world contexts, being a form of testing the training program's effectiveness and compliance of the acquired knowledge with the learning objectives.

Developing sophisticated skills and knowledge pertaining to competencies simultaneously from several different programming technologies is a challenging task. Its solution remains one of the most important educational issues, but is still little covered in the academic literature. This study emphasizes the conclusions previously made by other researchers. Sophisticated competencies and skills in programming and other IT careers can be learned more effectively at universities rather than in the workplace, requiring a shift in the learning paradigm [10, 40].

5. Conclusion

The design and management of databases is closely related to parallel computing. Such technologies are increasingly deeply integrated in the software companies' real-world contexts. Yet, academic research pays rather little attention to special teaching approaches in training personnel capable of immediately implementing such technologies in the workplace. Academic research focusing on developing programs or approaches to professional training in these areas in universities is relatively scarce. This study provides a theoretical background and practical approach to building parallel database programming skills by introducing software developers' best practices. The described approach was implemented by 64 third-year IT students of experimental group, which were involved in 3-month training program, building own personalized learning pathway when trying to solve 15 academic tasks. The tasks were developed by employees of software companies dealing with parallel databases. These students were also included into a group for assessment of the students' performance pertaining to the above software development tasks. Furthermore, to compare the effectiveness of the approach, the same tasks were given to 64 third-year students with a similar level of training, but who did not take part in the experiment (the control group). Some students in the experimental group were able to quickly improve their technology proficiency to an acceptable level. The control group did not achieve such a result. The Student's t-test showed that the differences in average experts' scores between the control and experimental groups were statistically significant. The research findings can be used to train students in parallel databases and set up connections with graduates' potential employers in relevant training areas.

Availability of data and materials. The data that support the findings of this study are available from the corresponding author upon reasonable request.

Competing interests. The authors declare that they have no competing interests.

Funding. The authors received no financial support for the research, authorship, and/or publication of this article.

References

1. L. Chen, *Flow Scheduling for Parallel Computing Applications in Datacenters*, PhD dissertation, Hong Kong University of Science and Technology, 2018.
2. S. Deshmukh, *Review of Research about Learning Analytics in Smart Learning Environments for Programming Education*, Master's thesis, Itä-Suomen yliopisto, 2021.
3. P. Kordjamshidi, D. Roth and K. Kersting, Systems AI: A declarative learning based programming perspective, in *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence*, IJCAI, pp 5464–5471, 2018.
4. Y. Yang, Q. Cao and H. Jiang, EdgeDB: an efficient time-series database for edge computing, *IEEE Access*, **7**, pp. 142295–14307, 2019.
5. J. Yin, D. Yuan, G. Wua, Z. Wang, Y. Zeng, J. Ling and Q. Lia, Construction of big database in the field of public security: a case study in Shenzhen, in *Proceedings of the 2nd International Conference on Green Energy, Environment and Sustainable Development (GEESD2021)*, IOS Press, Amsterdam, pp. 588–594, 2021.
6. J. Zhang, Y. Liu, K. Zhou, G. Li, Z. Xiao, B. Cheng, J. Xing, Y. Wang, T. Cheng, L. Liu, M. Ran and Z. Li, An end-to-end automatic cloud database tuning system using deep reinforcement learning, in *Proceedings of the 2019 International Conference on Management of Data*, ACM, New York, pp. 415–432, 2019.
7. X. Zhou, C. Chai, G. Li and J. Sun, Database meets artificial intelligence: a survey, *IEEE Trans. Knowl. Data. Eng.*, **34**(3), pp. 1096–1116, 2020.
8. S.W. Dietrich, Teaching introduction to programming languages with a database twist, *J. Comput. Sci. Coll.*, **36**(3), pp. 43–52, 2020.
9. T. D. J. O. Duraes, P. S. L. de Souza, G. Martins, D. J. Conte, N. G. Bachiega and S. M. Bruschi, Research on parallel computing teaching: state of the art and future directions, in *2020 IEEE Frontiers in Education Conference (FIE)*, IEEE, Uppsala, Sweden, pp. 1–9, 2020.
10. W. Gong, Database programming technology based on computer software engineering, in *Journal of Physics: Conference Series*, IOP Publishing, **2173**(1), p. 012073, 2022.
11. A. Fernando and L. Warusawithana, Database programming, in *Beginning Ballerina Programming*, Apress, Berkeley, CA, pp. 269–322, 2020.
12. H. Rahmalan, S. S. S. Ahmad and L. S. Affendey, Investigation on designing a fun and interactive learning approach for Database Programming subject according to students' preferences, in *Journal of Physics: Conference Series*, IOP Publishing, **1529**(2), p. 022076, 2020.
13. M. Serik, N. Karelkhan, A. Sadvakassova and G. Sarzhanova, Improvement of students' training in parallel and cloud computing, *Revista ESPACIOS*, **38**(60), pp. 1–3, 2017.

14. P. Yiatrou, *The Synthesis of a Unified Pedagogy for the Design and Evaluation of E-Learning Software for High-School Computing*, PhD dissertation, University of Central Lancashire, 2019.
15. A. Vasoya and N. Koli, A way to generate frequent pattern mining using vertical partitioning of data with parallel computing, *IJRASET*, **9**(1), pp. 191–197, 2021.
16. V. Garousi, G. Giray, E. Tuzun, C. Catal and M. Felderer, Closing the gap between software engineering education and industrial needs, *IEEE Software*, **37**(2), pp. 68–77, 2019.
17. B. Salah, M. H. Abidi, S. H. Mian, M. Krid, H. Alkhalefah and A. Abdo, Virtual reality-based engineering education to enhance manufacturing sustainability in industry 4.0, *Sustainability*, **11**(5), p. 1477, 2019.
18. V. Stuikys, R. Burbaite, G. Ziberkas and R. Kubiliunas, Development and evaluation of an approach for integrating data science concepts into high school STEM curriculum, *Int. J. Eng. Educ.*, **38**(3), pp. 756–773, 2022.
19. C. Rennick, G. Litster, A. Hurst, C. Hulls and S. Bedi, Characterizing engineering design activities using Jonassen's design theory of problem solving, *Int. J. Eng. Educ.*, **38**(1), pp. 56–66, 2021.
20. M. K. Shaikh, T. A. Jilani and K. Ahsan, Teaching computer science and computer engineering during COVID-19 lockdown at the Pakistani universities, *Int. J. Eng. Educ.*, **38**(2), pp. 322–334, 2022.
21. Z. Xiong and K. Xu, Lightweight job submission and file sharing schemes for a teaching ecosystem for parallel computing courses, *JAIHC*, in press, 2020.
22. E. Fernando, D. F. Murad and B. D. Wijanarko, Classification and advantages parallel computing in process computation: a systematic literature review, in *2018 International Conference on Computing, Engineering, and Design (ICCED)*, IEEE, Bangkok, Thailand, pp. 143–147, 2018.
23. R. Healey, S. Dowers, B. Gittings and M. J. Mineter, *Parallel Processing Algorithms for GIS*, CRC Press, Boca Raton, Florida, 2020.
24. L. H. Martins-Pacheco, N. D. Cruz Alves and C. G. V. Wangenheim, Educational practices in computational thinking: Assessment, pedagogical aspects, limits, and possibilities: A systematic mapping study, in *International Conference on Computer Supported Education*, Springer, Cham, pp. 442–466, 2019.
25. S. Han, Z. Li, P. Su, S. Li and Y. Wang, An empirical analysis of Neo-Piagetian theory in programming education, in *2021 International Conference on Culture-Oriented Science and Technology (ICCST)*, IEEE, Beijing, China, pp. 208–211, 2021.
26. F. Sanati, L. Soon and Y. Lin, Intelligent teaching and learning platform for introductory programming subjects, in *Proceedings of the 12th International Conference on Computer Modeling and Simulation*, ACM, New York, pp. 3–8, 2020.
27. M. R. Salaberry, A theoretical foundation for the development of pedagogical tasks in computer mediated communication, *CALICO J.*, **14**(1), pp. 5–34, 1996.
28. M. Serik and N. Karelkhan, Setting up and implementation of the parallel computing cluster in higher education, *Int. J. Emerg. Technol. Learn.*, **14**(6), pp. 4–15, 2019.
29. M. Serik and N. Karelkhan, Pedagogical foundations of learning parallel computing, *KazNU Journal. Ped. Sci. Series*, **61**(4), pp. 154–162, 2020.
30. J. A. Ruipérez-Valiente, P. J. Munoz-Merino, J. A. Gascón-Pinedo and C. D. Kloos, Scaling to massiveness with analyse: A learning analytics tool for open edx, *IEEE Trans. Hum. Mach.*, **47**(6), pp. 909–914, 2016.
31. Z. Shae and J. Tsai, Transform blockchain into distributed parallel computing architecture for precision medicine, in *2018 IEEE 38th international conference on distributed computing systems (ICDCS).* IEEE, Vienna, Austria, pp. 1290–1299, 2018.
32. F. F. Sharipov, T. Y. Krotenko and M. A. Dyakonova, Transdisciplinary strategy of continuing engineering education, in *Engineering Economics: Decisions and Solutions from Eurasian Perspective*, Springer International Publishing, pp. 480–488, 2021.
33. J. Chen, A. Kolmos and X. Du, Forms of implementation and challenges of PBL in engineering education: a review of literature, *Eur. J. Eng. Educ.*, **46**(1), pp. 90–115, 2021.
34. L. C. Cheng, W. Li and J. C. Tseng, Effects of an automated programming assessment system on the learning performances of experienced and novice learners, *Interact. Learn. Environ.*, in press, 2021.
35. A. Rabelo, L. C. G. Maia and F. S. Parreiras, Performance analysis of computer science students in programming learning, in *Anais do XXVI Workshop sobre Educação em Computação*, SBC, pp. 1–10, 2018.
36. Q. Tian, L. Wang, C. Tang, J. Zhang and J. Xu, International development trend of quantum computing research based on bibliometric analysis, *Sci. Focus*, **14**(6), pp. 1–9, 2020.
37. X. Wei, Q. Gu, Y. Luo and G. Chen, The reform of computer experiment teaching based on O2O model, *Comput. Appl. Eng. Educ.*, **27**(1), pp. 102–111, 2019.
38. N. C. Schaller and A. T. Kitchen, Experiences in teaching parallel computing – Five years later, *ACM SIGCSE Bull*, **27**(3), pp. 15–20, 1995.
39. G. R. Garay, A. Tchernykh, A. Y. Drozdov, S. N. Garichev, S. Nesmachnow and M. Torres-Martinez, Visualization of VHDL-based simulations as a pedagogical tool for supporting computer science education, *J. Comput. Sci.*, **36**, p. 100652, 2019.
40. A. Goncharow, A. Boekelheide, M. Mcquaigue, D. Burlinson, E. Saule, K. Subramanian and J. Payton, Classifying pedagogical material to improve adoption of parallel and distributed computing topics, in *2019 IEEE International Parallel And Distributed Processing Symposium Workshops (IPDPSW)*, IEEE, Rio de Janeiro, Brazil, pp. 312–319, 2019.

Weihong Zhao is a Master of International Business School in Huanghai University, Qingdao, China. His research interests include engineering software and information technologies.

Meruert Serik is a Doctor of Pedagogical Sciences and works as Professor of Department of Computer Science in L. N. Gumilyov Eurasian National University, Nur-Sultan, Kazakhstan. Research interests: information society, information systems.

Saule Zhumagulova is a doctoral student of Department of Computer Science in L. N. Gumilyov Eurasian National University, Nur-Sultan, Kazakhstan. She is interested in information systems and its management.