

Experiences in the Teaching of the DataFlow Paradigm: Lessons Learned

Nemanja Trifunovic, Milos Kotlar, Ognjen Andric, Petar Trifunovic, Sasa Stojanovic, Milos Cvetanovic, Zaharije Radivojevic, Marija Punt, Nenad Korolija, and Veljko Milutinovic

Abstract—The DataFlow paradigm has been existing for a long time, but good enough enabling technologies have been made available only very recently. The teaching of a different from the mainstream paradigm, such as the DataFlow paradigm, encounters a few problems. If a new paradigm is presented among experienced professionals, the presentation is often received with resistance. If a new paradigm is presented before students, the presentation is always received with enthusiasm, especially if tools and libraries that support the paradigm do exist. In order to overcome the problem with introducing the new paradigm, a series of courses was organized for Universities in the USA and Europe, to teach the DataFlow paradigm, using the Maxeler approach as the baseline for practical work. The course is organized in three different levels: (A) full-hour presentation that includes introducing the programming model with hands-on examples, which is most appropriate for research groups at Universities (B) a certified full-day short-course that students could use as an extension to diploma, (C) full-semester course for credits that is part of graduation requirement. This paper discusses different methods and their results, which could be used in order to efficiently introduce any different from the mainstream paradigm. The results presented in this paper are for 10 universities in the group A, 10 in the group B, and 10 in the group C. Where appropriate, the findings are backed with statistics data.

Key words — big data, dataflow, paradigm shift, sea of gates, teaching

I. INTRODUCTION

THE best way to introduce the problem treated in this article is to quote Schopenhauer [1]: "Change passes through three stages. First, it is ridiculed. Second, it is violently opposed. Third, it is accepted as self-evident." In other words, professionals in a specific field refuse to accept a new paradigm and apply all three stages until the paradigm becomes widespread.

The main problem is how to pass the first two phases as

Paper received April 20, 2018; revised August 14, 2018; accepted September 10, 2018. Date of publication December 25, 2018. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Miroslav Lutovac.

This paper is a revised and expanded version of the paper presented at the 25th Telecommunications Forum TELFOR 2017 [17].

Nemanja Trifunovic(email: nemanja@maxeler.com), Milos Kotlar (email: kotlarmilos@gmail.com), Ognjen Andric (email: ognjen.andric91@gmail.com), Petar Trifunovic (email: petart95@gmail.com), Sasa Stojanovic (email: stojsasa@etf.bg.ac.rs), Milos Cvetanovic (email: cmilos@etf.bg.ac.rs), Zaharije Radivojevic (email: zaki@etf.rs), Marija Punt (email: maki@etf.rs), Nenad Korolija (email: nenadko@etf.rs), Veljko Milutinovic (email: vm@etf.rs), Elektrotehnički fakultet u Beogradu, Bulevar kralja Aleksandra 73, 11120 Beograd, Srbija.

fast as possible and with the least possible damage.

The resistance to paradigm shift is pronounced much more if the previous paradigm was lasting too long and was spanning the entire lifetime of one or two generations of professionals. An example of such a paradigm shift is multi-scale dataflow computing in general, and its various mutations in specific. The previous paradigm, based on the von Neumann approach to computing, was initiated in the 40's of the last century.

What a new paradigm shift offers in terms of speedup, power, size, and other benefits often gets ignored, because the fear related to the change of the way of thinking and acting typically prevails. The focus in this article is on educational approaches, tools, and examples that could be used in the process of introducing a new paradigm.

II. PROBLEM STATEMENT

The major problem is how to convince decision makers to adopt a specific new course, and how to organize the presentation layer for students, for maximal effectiveness, in terms of minimal time till adoption of a new paradigm. An example could be professional precision combined with anecdotes tuned to the mind types of typical students. This effort includes publishing, too.

The second major problem is to develop tools that make the life of a student easy, so his/her thinking on higher levels of abstraction could be easily ported to the actual code.

The third major problem is to develop a set of ready-to-use examples, which could be reused, with a smaller or a larger effort; it is important that these examples span a wide area of possible applications, so each student could find his/her interests properly covered.

The problem is important because the rate of acceptance of effective paradigm shifts has a direct impact on the economy of the nation or the world. As an example, the survey [16] indicates that computer networks need better performance in future exascale computers, which leads to paradigm shifts.

The problem will grow over time, since the professionals get exposed to more and more novelties, and it has become difficult to figure out which one is the appropriate one for their use.

III. EXISTING APPROACHES

In the domain of the educational methodologies used to introduce new paradigms, the most relevant are the experiences of OpenCourseWare [2], and for that reason, the next subsection includes the experiences of various

OpenCourseWare efforts, together with their short descriptions, only as much as needed for the better understanding of the approach described in this article.

In the domain of tools used to teach new paradigms, a short survey of such tools is given in subsection B.

In the domain of libraries of examples that support new paradigms, a short survey is given in subsection C.

A. *OpenCourseWare Experiences with Introduction of New Paradigms*

TheOpenCourseWare project, including the origin of the concept at the Massachusetts Institute of Technology (MIT) was spread fast into the global educational community [3]. This project covers many courses from different areas of research, such as medicine, computer science, civil engineering, and many others. Nowadays, there are a lot of open online courses at the best universities in the world, but none of them sheds light on paradigm shift methodologies.

B. *Existing Tools for Teaching New Paradigms*

In the open literature there are a lot of tools that are used for teaching, such as ADVICE [4] that helps students to bridge the gap between database management system (DBMS) theory and practice, and SLEEP [5] that helps students to bridge the gap between theory and practice in the domain of Computer Architecture and Organization simulator design. None of them includes a construct for paradigm shift teaching.

C. *Existing Libraries that Support New Paradigms*

Existing libraries are important in new paradigms, especially if they contain ready-to-use examples. This approach allows students to become familiar with new paradigms. If existing libraries cover all major concepts of a new paradigm, a student could find his/her interests properly covered.

IV. ESSENCE OF THE PROPOSED APPROACH

The proposed approach implies activities in three different domains: (a) Setting short introductory lectures at universities worldwide that inform local educators about the essence and advantages of the new paradigm in computing (multi-scale dataflow), which helps and motivate them to accept short and regular courses at their universities, so students get exposure; this activity is followed with short courses and full courses, as well as the related publishing activities; (b) Building tools to support teaching, i.e., helping educators to understand the paradigm, and helping students to master the issues; and (c) Creating libraries to support the dissemination process, by including ready-to-use examples.

In the first domain we created some experiences at various universities worldwide. We divide them into three groups: (A) Where only introductory talks were delivered; (B) Where both, introductory lectures and short courses were delivered; (C) Where the delivery of introductory lectures and short courses was followed by teaching in courses from which students could get credit.

In the part on Education (Tutorials), Tools, and Examples, it will be specified, what are the constructs tuned to more effective teaching of the paradigm-shifting issues.

V. CONDITIONS AND ASSUMPTIONS

The experiences presented here come from universities in 5 different geographical locations: 2 from the East Coast USA, 2 from the MidWest USA, 2 from the continental parts of EU, 2 from universities in Mediterranean countries, and 2 from ex-Yu universities:

A. Columbia, NYU, Michigan, Illinois, Stuttgart, Heidelberg, Bogazici, Ankara, Podgorica, Skopje

B. Princeton, Temple, Ohio State, CMU, Vienna Poli, Vienna Uni, Barcelona, Valencia, Novi Sad, Kragujevac

C. MIT, Harvard, Purdue, Indiana@Bloomington, FRI at Ljubljana, Koper, Siena, Salerno, MATF.R at Belgrade, MATF.I at Belgrade

In other words, at the universities in group A, only the introductory lecture was delivered; at the universities in group B, both introductory lectures and short 1-day courses were delivered; in group C, after the introductory lecture and a short course, students also had a chance to obtain credit needed for graduation.

VI. THEORETICAL ISSUES

All course materials could be found on the web [6]. The full-blown course was organized as a series of PowerPoint presentations (red bullets) followed by reading materials (blue bullets). The first PowerPoint presentation was used for the 1-hour introductory lecture; that one and a subset of all was used for the 1-day short course. The entire set of slides was used for the course for credit.

If a student attended the full course and answered properly all the exam questions, he/she would obtain 0 credits and a certificate of successful completion. If one homework is successfully completed (implementing a math algorithm), the student would deserve one credit. If one more is completed, related to image processing, the student would deserve two credits. For three credits, the student must complete one more homework assignment successfully, in one of the following fields: Machine Learning, Genomics, Finances, Oil and Gas.

The lecturing is supported by two series of books: One by Springer [7]-[9], on educational issues, and the other one by Elsevier [10]-[12], on research issues.

The one-hour introductory lecture sheds light on the roots of the invention, on its essence, and on the major achievements in the domains of speed, power, size, and precision. The idea is that after exposure to such a lecture, the decision makers of universities decide to include the subject into informative or regular teaching at their university.

The one-day short course teaches the basics and ensures that students can solve small problems by themselves, without much help from others, who are more experienced.

The full-semester course enables students to solve the most sophisticated problems on their own. For that to happen, tutorials, tools, and examples were developed.

VII. PRACTICAL ISSUES

This section sheds light on the efforts related to the creation of tutorials and other educational materials, on the creation of tools, and on the creation of examples organized in a library called AppGallery.Maxeler.com (in each part,

special attention was dedicated to the components that are aiming at faster adoption of a new paradigm).

A. Tutorials

In the core of the dataflow programming is the concept of programming in space, arranging operations in two dimensions, and making a program that configures the underlying hardware in such a way to minimize the data movement and thus minimize the power usage and to maximize the hardware utilization and thus maximise the achieved compute density [13].

MaxelerWebIDE is a development tool aiming students to build dataflow applications, as shown in Fig. 1. It contains templates that are used as a basis for new applications. This tool represents a construct for paradigm shift teaching.



Fig. 1. The WebIDE development tool.

B. Tools

There are several useful tools to aid developer write the highest performance dataflow applications [14].

- Maxtop is a command line tool that is used to monitor the state of the dataflow system.
- MaxIDE is an integrated development environment for programming dataflow engines.
- MaxPower is a standard library written in Maxeler Java Language (MaxJ) containing a lot of pre-made components for writing dataflow programs packaged in a reusable way.
- Monitoring dashboard is a service for monitoring dataflow systems. While Maxtop is a command line tool, monitoring dashboard provides the modern web user interface that can be used to check the current state of the dataflow system from the web browser.
- MaxDebug tool is a command line utility that allows inspection of debug snapshots that are optionally produced by the Simple Live CPU Interface (SLiC) applications as their actions are completed. The tool can obtain the status of the DataFlow Engine (DFE) while it is running or after the execution has completed.
- Sliccompile generates SLiC skins from the .max files. Sliccompile can generate SLiC interfaces for C (default interface), Python, MatLab and R.
- MaxSkins enable easy and seamless access to the DFEs from the remote machines over the TCP/IP network from various programming languages.
- MaxOrchestrator is a software daemon that manages the cluster of machines that are having DFEs

exposed over the InfiniBand (MPC-X 1U nodes) to the rest of the compute nodes in the cluster connected to the same InfiniBand network.

All the above mentioned tools together allow a new paradigm to be introduced faster and easier.

C. AppGallery

The AppGallery contains more than 60 applications and application tools [15], as shown in Fig. 2. Those applications and tools are divided into six main categories: Data Analytics, Engineering, Low Latency Transaction Processing, Networking, Science, and Security.

Available applications can be filtered using sixteen characteristics:

1. Source of the CPU code available
2. Source code written in MaxJ
3. Complete source code available
4. App use case documentation available
5. App technology documentation available
6. The App has a GUI
7. There is a link to a video about the App available
8. The source code of the original software application available
9. Another version of the original source code split into DFE and CPU part available
10. The author might be interested in providing a commercial license
11. The App has a single DFE interface
12. The App has a DFE interface
13. The App has a multi DFE interface
14. The App supports MAX3 DFEs
15. The App supports MAX4 DFEs (allows choosing one of four platforms - MAIA, ISCA, GALAVA, and CORIA)
16. The App supports JDFE generations cards.

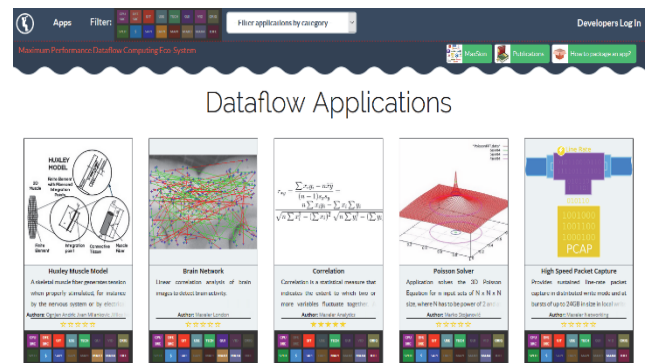


Fig. 2. The AppGallery.

The AppGallery together with Maxeler WebIDE development tool represents an efficient construct for paradigm shift.

VIII. STATISTICS

The presentation to follow stresses the numeric that tell about the enthusiasm for adoption of the new paradigm, as well as about the problems encountered on the way.

A. The Attendance Record

Table 1 presents the attendance log at all universities mentioned above, year after year, for the total of close to about 800 students.

TABLE 1: THE ATTENDANCE LOG AT ALL UNIVERSITIES

Universities	I	II	III	IV
Harvard CSE program	175	220	-	-
MIT MEDIA lab	14	12	6	11
Purdue CSE	40	24	13	18
Indiana University at Bloomington	14	8	11	9
Ljubljana in Slovenia, FRI	24	11	-	-
Koper in Slovenia	34	32	-	-
Siena in Italy	22	-	-	-
Salerno	17	-	-	-
Belgrade, MATH,program R	3	4	5	13
Belgrade, MATH, program I	36	17	26	14

B. The Success Record

This section presents how successful the students were as far as achieving speedups on their homework assignments, and why these speedups were not as high as permitted by potentials.

Most of the students were able to compile and run successfully (over 90%), only a smaller percentage was able to generate speedup over I7 (about 20%) and very few were able to generate a considerable speedup (about 5%). The speedups were not high, since one semester, together with other student loads, was not enough for the acquisition of all necessary skills.

C. The Internship Record

This part tells about the achievements of those who decided to do their internship at Maxeler; only for those who were able to achieve a considerable speedup. For each one, a topic was presented, the way the problem was attacked.

The main result of university teaching was that students were able to understand the essence of the approach and to generate enthusiasm for the approach. They would become effective programmers of the paradigm, only after they spend some time on internships. Topics would include: Math, Image processing, Machine learning, and Computational physics.

D. The AppGallery Record

This part tells about the major successes of those who decided to work permanently for Maxeler, and what was the essence of their AppGallery contribution.

What the students were able to achieve, is best visible at appgallery.maxeler.com (if a work was done by an employee, the company name appears; if the work was done by a student, a personal name appears). The essence is visible through accompanying PowerPoint presentations.

E. The Q+A Record on the Higher Levels of Abstraction

This section tells about the most frequent essential questions that young professionals had during all the above-mentioned phases of their development.

The concept of optimization was not well understood in this environment at the beginning. In control flow, optimization means decreasing the number of iterations of an algorithm, and/or decreasing the length of one iteration. In dataflow, optimization means decreasing the length of edges and decreasing the execution graph deteriorations due to the mapping to some inflexible hardware. Also, the concept of programming in space, rather than programming in time, had to be mastered.

F. The Q+A Record on the Lowest Implementation Level

This section sheds light on the most frequent implementation-related questions that helped achieve high quality performance results. These questions were the major input for creation of the next generation of tools.

Most of the questions were related to the problems with compilation and running. Students typically do proper job on the kernel programming side, but they forget to do proper modifications on the host code side.

IX. CONCLUSIONS

This paper presents an original and effective approach that enables teaching of issues related to a different paradigm, with existing tools and libraries that support the teaching process. The multi-scale DataFlow paradigm mentioned in this paper is based on the Maxeler machines and related infrastructure.

The results presented in this work are of potential benefit to others who are facing the same problem: resistance to the adoption of a new paradigm, although the paradigm demonstrates superiority and has great potentials for further improvements.

The major problem, among the newly open problems pointed to by this research, refers to the development of the generalized methodology that could be used to overcome the problem with introducing a new paradigm. This article sheds light on some of the starting points of the development of such a methodology.

REFERENCES

- [1] Schopenhauer, Arthur. *Arthur Schopenhauer: The World as Will and Presentation: Volume I*. Routledge, 2016.
- [2] Abelson, Hal. "The creation of OpenCourseWare at MIT." *Journal of Science Education and Technology* 17.2 (2008): 164-174.
- [3] Carson, Steve. "The unwallied garden: growth of the OpenCourseWare Consortium, 2001–2008." *Open Learning* 24.1 (2009): 23-29.
- [4] Cvetanovic, Miloš, et al. "ADVICE - Educational system for teaching database courses." *IEEE Transactions on Education* 54.3 (2011): 398-409.
- [5] Radivojevic, Zaharije, Milos Cvetanovic, and Jovan Đorđević. "Design of the simulator for teaching computer architecture and organization." *Engineering of Computer Based Systems (ECBS-EERC), 2011 2nd Eastern European Regional Conference on the*. IEEE, 2011.
- [6] <http://home.etf.rs/~vm/os/index.html>, [Accessed 01.10.2017]
- [7] Milutinović, Veljko, et al. *Guide to dataflow supercomputing*. Springer, 2015.
- [8] Milutinovic, Veljko, et al. *DataFlow Supercomputing Essentials: Algorithms, Applications and Implementations*. Springer, 2017.
- [9] Milutinovic, Veljko, et al. *DataFlow Supercomputing Essentials: Research, Development and Education*. Springer, 2017.
- [10] Hurson, A., Milutinovic, V., *Advances in Computers*, Elsevier, vol. 96, 2015.
- [11] Hurson, A., Milutinovic, V., *Advances in Computers*, Elsevier, vol. 104, 2017.
- [12] Hurson, A., Milutinovic, V., *Advances in Computers*, Elsevier, vol. 106, 2017.
- [13] Trifunovic, N., Milutinovic, V. et al, "Paradigm Shift in SuperComputing: DataFlow vs ControlFlow," *Journal of Big Data*, 2015.
- [14] Flynn, M., Mencer, O., Milutinovic, V., et al, Moving from PetaFlops to PetaData, *Communications of the ACM*, 2013.
- [15] Trifunovic, N., Milutinovic, V. et al, "The AppGallery.Maxeler.com for BigData SuperComputing," *Journal of Big Data*, Springer, 2016.
- [16] Trobec, Roman, et al. "Interconnection networks in petascale computer systems: A survey." *ACM Computing Surveys (CSUR)* 49.3 (2016): 44.
- [17] N. Trifunovic et al., "Experiences in the teaching of a new computing paradigm: DataFlow supercomputing," *2017 25th Telecommunication Forum (TELFOR)*, Belgrade, 2017.