BalticLSC

BalticLSC Software Technical Communications Report

Results of the BalticLSC Software Technical Communication

Version 1.00

Interreg
Baltic Sea Region

EUROPEAN UNION

EUROPEAN REGIONAL DEVELOPMENT FUND

Priority 1: Innovation

Warsaw University of Technology, Poland
RISE Research Institutes of Sweden AB, Sweden
Institute of Mathematics and Computer Science, University of Latvia, Latvia
EurA AG, Germany
Municipality of Vejle, Denmark
Lithuanian Innovation Center, Lithuania
Machine Technology Center Turku Ltd., Finland
Tartu Science Park Foundation, Estonia

BalticLSC Software Technical Communications Report

Results of the BalticLSC Software Technical Communication

Work package	WP5
Task id	5.6
Document number	O5.6
Document type	Output
Title	BalticLSC Software Technical Communications Report
Subtitle	Results of the BalticLSC Software Technical Communication
Author(s)	Krzysztof Marek (WUT), Agris Šostaks (IMCS)
Reviewer(s)	Edgars Celms (IMCS)
Accepting	Michał Śmiałek
Version	1.00
Status	Final version

Interreg
Baltic Sea Region

EUROPEAN UNION

EUROPEAN REGIONAL DEVELOPMENT FUND

History of changes

Date	Ver.	Author(s)	Change description
18.12.2021	0.1	Krzysztof Marek (WUT)	Information about WUT technical communication
21.12.2021	0.2	Agris Šostaks (IMCS)	Added info on paper
28.12.2021	0.3	Edgars Celms (IMCS)	Review
29.12.2021	1.0	Krzysztof Marek (WUT)	Final version

Table of contents

1. Introduction

1.1 Objectives and scope

The objective of this document is to summarize all the technical communication of the BalticLSC Software. By technical communication we understand scientific, reviewed papers, white papers, books, articles and scientific conferences. This does not include standard BalticLSC communication which was a part of GoA2.4 and has been described in O2.4.

1.2 Relations to Other Documents

This document remains in a very close relation with the joined outputs O5.2 – BalticLSC Admin Tool Technical Documentation, O5.3 – Computation Application Language Manual, and O5.4 – BalticLSC Computation Tool Technical Documentation which are the design artifacts of different components of the BalticLSC Software which were the base for the technical communication.

The second important document is the O6.2 the implemented BalticLSC Software which has been described and promoted during the technical communication.

1.3 Intended Audience and Usage Guidelines

This document is intended for everyone interested in learning about the innovative solutions developed during the BalticLSC project for the BalticLSC Software. This document summarizes these communication efforts and describes created artifacts in form of scientific papers, books, articles and conference presentations.

All technical communication artefacts that can be publicly presented are available at: [Project Outputs - BalticLSC](https://www.balticlsc.eu/downloads/project-outputs/) (https://www.balticlsc.eu/downloads/project-outputs/)

2. BalticLSC Software papers, books and articles

The BalticLSC Software and its parts have been presented in multiple scientific papers and articles.

2.1 Towards a Unified Requirements Model for Distributed High Performance Computing

Authored by: Michał Śmiałek (WUT), Kamil Rybiński (WUT), Radosław Roszczyk (WUT) and Krzysztof Marek (WUT)

Description: This chapter in a peer reviewed book “Data-Centric Business and Applications” edited by Aneta Poniszewska-Marańda, Natalia Kryvinska, Stanisław Jarząbek, and Lech Madeyski proposes a unified requirements model for distributed High Performance Computing based on the requirements for the BalticLSC Software.

Abstract: High Performance Computing (HPC) consists in development and execution of sophisticated computation applications, developed by highly skilled IT personnel. Several past studies report significant problems with applying HPC in industry practice. This is caused by lack of necessary IT skills in developing highly parallelised and distributed computation software. This calls for new methods to reduce software development effort when constructing new computation applications. In this paper we propose a generic requirements model consisting of a conceptual domain specification, unified domain vocabulary and use-case-based functional requirements. Vocabulary definition provides detailed clarifications of HPC fundamental component elements and their role in the system. Further we address security issues by providing transparency principles for HPC. We also propose a research agenda that leads to the creation of a model-based software development system dedicated to building Distributed HPC applications at a high level of abstraction, with the object of making HPC more available for smaller institutions.

Published: 15 December 2019

DOI: 10.1007/978-3-030-34706-2_1

Downloads: 400 at Springer, 50 at ResearchGate (as of December 2021)

Citation: 1 in a scientific article from 2021 (as of December 2021)

2.2 BalticLSC: A low-code HPC platform for small and medium research teams

Authored by: Radosław Roszczyk (WUT), Marek Wdowiak (WUT), Michał Śmiałek (WUT), Kamil Rybiński (WUT) and Krzysztof Marek (WUT)

Description: The BalticLSC Environment with the special focus on the CAL language has been presented at the international Symposium on Visual Languages and Human-Centric Computing (VL/HCC) in St Louis, USA. The BalticLSC and CAL have been presented during an online poster session where interested participants could come to BalticLSC virtual booth and learn about the BalticLSC solutions.

Abstract: Research teams often face problems with insufficient resources to develop and execute their computation algorithms. This is due to limited access to professional programmers and High Performance Computing centres. This paper presents a platform that uses a new visual language to solve these problems. It uses high-level abstractions to define the flow of data and the execution of computation modules in a distributed computation environment. The runtime system follows the rules of low-code, where the development is on-line and highly simplifies the deployment of apps. This way, even small research teams can easily gain access to high-end computation resources and develop and share their applications.

Date of Conference: 10-13 October 2021

Published: 29 October 2021

DOI: 10.1109/VL/HCC51201.2021.9576305

Downloads: over 30 at IEEE Xplore (as of December 2021)

2.3 BalticLSC: Low-code Software Development Platform for Large Scale Computations

Authored by: Krzysztof Marek (WUT), Michał Śmiałek (WUT), Kamil Rybiński (WUT), Radosław Roszczyk (WUT) and Marek Wdowiak (WUT)

Description: The BalticLSC Environment and the low-code software development process made available via the BalticLSC Software has been described in an article in an international peer reviewed journal "Computing and Informatics".

Abstract: In modern times, innovation often requires performing complex computations in a short amount of time. However, for many small organisations and freelance innovators, large-scale computations remain beyond reach because of the small accessibility of computation resources and the lack of knowledge required to use them efficiently. The BalticLSC Platform is a software development and computing environment created to address this issue. This paper presents the associated software development process. The platform users can perform advanced computations using ready applications or develop new applications quickly from available components. This can be done using a visual notation called the Computation Application Language (CAL). CAL programs are developed in a dedicated online editor, through selecting and connecting reusable computation modules. If a required module is missing, it can be quickly created by encapsulating code inside a standardised container. The platform's ultimate goal is to relieve the developers from the need to understand the complexity of the distributed parallel computation environment. The platform was implemented in the form of an online software development portal. Validation of the platform consisted in the development of applications and modules by students not experienced in programming. The results of this validation acknowledge the required platform's characteristics.

Published: 14.12.2021

DOI: 10.31577/cai_2021_4_734

2.4 Data Flow Driven Low-Code Language for Distributed Computations

Authored by: Kamil Rybiński (WUT), Michał Śmiałek (WUT), Marek Wdowiak (WUT), Agris Sostaks (IMCS) and Radosław Roszczyk (WUT)

Description: This article presents the innovative Computation Application Language developed for the purpose of BalticLSC Software to allow for easy Computation Application Development without the need to write code.

Published: The article has been already written and is during the process of publishing

2.5 Experience in Developing Computation Applications Using the BalticLSC System

Authored by: Adam Gawieńczuk (WUT), Jena Sanjeeta Smruti (WUT) and Robert Mazurek (WUT)

Description: This article presents the validation of Computation Application Development Process performed by creating a new application by people without significant experience in software development and not experience in large-scale computations. The authors developed their own Computation Modules in C# and used them in a new Computation Application defined in CAL.

Abstract: BalticLSC is a new platform for large-scale computing, and it is still in development. In this document, we analyzed the complexity of creating new modules and applications and publishing them on the BalticLSC platform from the perspective of software development students with little experience. It also includes some instructions on how to make said applications.

Published: The article has been already written and is during the process of publishing

2.6 Estimation of Execution Time for Computing Tasks

Authored by**:** Jan Bielecki (WUT), Michał Śmiałek (WUT)

Abstract: The aim of this work is to estimate the execution time of data processing tasks (specific executions of a program or an algorithm) before their execution. The paper focuses on the estimation of the average-case execution time (ACET). This metric can be used to predict the approximate cost of computations, e.g. when resource consumption in a High-Performance Computing system has to be known in advance. The presented approach proposes to create machine learning models using historical data. The models use program metadata (properties of input data) and parameters of the runtime environment as their explanatory variables. Moreover, the set of these variable scan be easily expanded with additional parameters of the specific programs. The response variable of the model is the execution time. The models have been validated within a Large-Scale Computing system that allows for a unified treatment of programs as computation modules. We present the process of training and validation for several different computation modules and discuss the suitability of the proposed models for ACET estimation in various computing environments.

Published: The article has been already written and is during the process of publishing.

2.7 Integration of Graphical DSL Editor: BalticLSC Use case

Authored by: Edgars Celms (IMCS), Sergejs Rikačovs (IMCS), Agris Šostaks (IMCS), Artūrs Sproģis (IMCS)

Description: Graphical DSL Building Frameworks, e.g., Sirius, offer rich possibilities to develop, maintain and deploy the graphical DSL editors. However, these editors are hard to integrate into the existing software systems because frameworks come with their own ecosystem - repositories, technologies, and restrictions. We offer lightweight JavaScript library which can be easily configured, extended, and integrated into any popular web-based framework. We build the graphical editor for the Computation Application Language within Interreg BalticLSC project. We use this case study to show that our approach allows more flexibility and more options for easy integration of graphical editors into the external software systems.

Published: The article has been already written and is during the process of publishing.

EUROPEAN REGIONAL DEVELOPMENT FUND EUROPEAN UNION

3. BalticLSC Software scientific conference

The BalticLSC Software has been also presented at scientific conferences that don't publish scientific proceeding (presentations at conference with proceedings have been already described in the previous chapter).

3.1 XXII KKIO Software Engineering Conference

Presenters: Michał Śmiałek (WUT) and Krzysztof Marek (WUT)

Description: The BalticLSC Environment has been presented during workshops organized during the KKIO conference. During the interactive workshops participants learned about the BalticLSC project and could develop their own image processing Computation Application using the BalticLSC Prototype available at www.balticlsc.eu.

Abstract: A practical workshop dedicated to developing low code computation applications within the Baltic Large-Scale Computing Platform (www.balticlsc.eu). After a brief introduction to the BalticLSC Platform, the participants will develop and run their own low code application tasks. Every participant will gain individual remote access to the BalticLSC Prototype Platform (subject to availability) and will be able to perform computations on their own. The only requirement to participate in the workshop is to have a computer with internet access and an up-to-date web browser

Conference date: 21-22.09.2021

Workshop agenda:

1. Introduction to the BalticLSC Computation Platform – goals and possible applications
2. Architecture of the BalticLSC Computation Platform
3. Introduction to CAL – Low-code visual language
4. Practical workshop – running an application and managing your data on the Platform
5. Practical workshop – developing and running a low-code image processing application in CAL
6. Questions and discussion

Workshops participants: around 40

3.2 VL/HCC 2021 – The Symposium on Visual Languages and Human-Centric Computing

Presenters: Radosław Roszczyk (WUT), Michał Śmiałek (WUT), Kamil Rybiński (WUT) and Krzysztof Marek (WUT)

Description: The BalticLSC Environment has been presented during online conference (this year hosted in the USA, with multiple EU participants). Because of the online form of the conference, firstly the BalticLSC Environment has been presented in a form of a short video recorded by Michał Śmiałek specifically for this event. After a few presentations the conference participants moved to the online discussion hall using Gather Town. During that session all conference participants could move between different zones and discuss the presentations with the presenters. During that part of the conference we presented the BalticLSC Environment in practice to anyone interested and answered questions regarding the solution. The short article written for this conference has been described in Section 2.2.

Conference date: 10-13.10.2021

4. Conclusions

The BalticLSC Software and it’s different parts have been communicated in multiple official technical communication channels. Due to the specificity of technical communication in which every paper has to be peer reviewed by multiple experts from the field the process takes a lot of time. The technical communication was possible mainly at the end of the project when the BalticLSC Software Prototype has been developed. Because of this the papers have been only recently published (or are waiting to be published by the publishers) and measuring their impact by citations and/or number of reads is impossible. The only technical communication paper published at the beginning of the BalticLSC Project (that is after a year since it started) has been already cited and according to the statistics we have access to read by at least 450 people most of them being experts and scientists. This shown interest in the BalticLSC Software technical communication, however its real impact can be measured in few years.