CERCIRAS Training School 2023: Advanced Topics in Resource-Aware Computing

Technical Program

- **Venue**: Riga Technical University, Kipsala campus, @ Riga, LV
- **Dates**: from Monday, 4 September until Saturday, 9 September
- **Capacity**: up to 26 trainees accommodated in single rooms at Kipsala Kampus and nearby facilities (on a first-come-first served basis), with CERCIRAS reimbursement for travel and subsistence expenses after registration in CERCIRAS' e-cost platform.
- **Program synopsis**: see table below, where the X.Y topic identifier denotes the topic number X (the school features **four** lecture topics) and the slot Y in each lecture series (each lecture series has **four** 90-minute slots, which alternate theory and hands-on practice).
- **Optional exams**: for trainees who need a "pass mark" from the school, on-site exams will be organized before the end of the school by the corresponding instructors.
- **Other program features**:
  - The trainees will present their current research or professional work and project highlights in a *poster session* that will take place in the evening of Tuesday. Each trainee will have 5 minutes of airtime to present themselves *using one single presentation slide* from their own laptop.
  - A *diversion excursion* will take place in the afternoon of Friday (to be confirmed).
  - An *exchange-and-discussion session* will take place, in the evening of Friday, between the trainers and the trainees on the actual learning outcome of the School and the research prospects that emanate from the lecture topics addressed in it.

<table>
<tr><th></th><th>Monday</th><th>Tuesday</th><th>Wednesday</th><th>Thursday</th><th>Friday</th><th>Saturday</th></tr>
<tr><td>09:00 – 10:30</td><td rowspan="2"></td><td>Topic 3.1</td><td>Topic 3.2</td><td>Topic 3.3</td><td>Topic 3.4</td><td rowspan="2">Closing and Departures</td></tr>
<tr><td>10:30 – 11:00</td><td colspan="4">Morning Break</td></tr>
<tr><td>11:00 – 12:30</td><td>Welcome and Opening</td><td>Topic 4.1</td><td>Topic 4.2</td><td>Topic 4.3</td><td>Topic 4.4</td></tr>
<tr><td>12:30 – 14:00</td><td colspan="5">Joint lunch</td><td rowspan="8"></td></tr>
<tr><td>14:00 – 15:30</td><td>Topic 1.1</td><td>Topic 1.2</td><td>Topic 1.3</td><td>Topic 1.4</td><td rowspan="3">Excursion</td></tr>
<tr><td>15:30 – 16:00</td><td colspan="4">Afternoon Break</td></tr>
<tr><td>16:00 – 17:30</td><td>Topic 2.1</td><td>Topic 2.2</td><td>Topic 2.3</td><td>Topic 2.4</td></tr>
<tr><td></td><td colspan="5">Free time (for study and interaction with instructors)</td></tr>
<tr><td></td><td colspan="5">Joint dinner</td></tr>
<tr><td>19:45-20:45</td><td></td><td>Poster Session</td><td></td><td></td><td>Future Research Prospects</td></tr>
</table>

Table of Contents

Table of Contents

Lecture Series 1:

Instructor: Nahuel Palumbo, Inria, Lille, France

- Nahuel Palumbo was teaching programming at different national universities in Argentina for 10 years. These lectures include topics related to object-oriented programming and software design. He also worked in Uqbar and Sadosky, two Argentinian foundations where we research and develop tools for teaching and learning programming. In this context, he has been developing programming languages for the last 6 years. Presently, he is in the second year of his PhD. His work is centred on building Pharo, an object-oriented programming language dynamically typed, specifically in the Virtual Machine (VM). His thesis is related to optimising compilers, but he loves to learn and improve other components of the VM too, like Garbage Collectors.

Outline

Automatic memory management is often supported by Garbage Collectors (GC). GCs have to support programs that manipulate different types of data in different manners. Such data must be allocated in dynamic memory (the heap), to be easily accessible when the application needs it and to be freed when no longer needed. This dynamics is termed “allocation pattern”. GCs try to predict the allocation pattern of running applications for better performance. So, they implement parametric algorithms to adapt their complex behaviour to each specific application. There are reports about how garbage collection impacts application performance if not tuned properly, but most developers do not know how the GC works with their programs.

In this lecture series, we illustrate how Garbage Collectors work in the context of object-oriented programs. We use the Pharo language to profile the program allocations and analyse profile data. We will analyse applications with different allocation patterns and identify possible causes of performance degradation (pathological cases). We use this information for tuning GC parameters and improving the application’s performance.

Lab requirements

The practical part of the lectures will require pre-installed Git and Pharo (https://pharo.org) for cloning and profiling programs. The tools for the analysis of profile data are also developed in Pharo.

Basic knowledge of programming is desired. Object-oriented programming will be introduced on the first day of the lecture series but precursor knowledge is desired.

Slot 1 (Lecture): Introduction to Pharo and Object-Oriented Programming

This lecture will provide a basic understanding of object-oriented programming (object, message, polymorphism, block closures) using Pharo. We will present the syntax of the language and explore the Pharo environment. We will use the tools (testing, inspectors, profilers) with simple programs. We expect the required software to already be installed in the participants’ devices. Missing installations will be performed live, if needed.

Slot 2 (Lecture): Introduction to Garbage Collection

The lecture will introduce Garbage Collectors (GC) in the context of object-oriented programs. We will present the main contribution of GCs and some known algorithms. We will describe how the GC is implemented inside the Pharo Virtual Machine and the exposed parameters for tuning it on running applications.

Slot 3 (Lecture and Practicals): Analysis of memory allocation patterns of trial applications

We will profile the Garbage Collector events for different application executions. We will analyse profile data based on different charts for understanding how the GC interacts with running applications. We will identify pathological cases, and the possible causes of the performance degradation. We will tune the GC to achieve better application performance. Work in this lecture will be carried out collaboratively by the lecturer and the trainees.

Slot 4 (Practicals): Profiling Garbage Collector events on running applications

The lecturer will single out one application case to analyse the impact of the GC on its performance. For pathological cases, the trainees will be guided to analyse the captured data, to identify possible causes of performance degradation and to understand how to tune the GC for solving them. They will identify the allocation patterns and understand whether and why the default values in the GC parameters perform good or bad.

Lecture Series 2: An introduction to Structured Parallel Programming using MPI Collectives

Instructor: Horacio Gonzalez-Velez, National College of Ireland, IE

- With over twenty years of HPC experience in industry and academia, Horacio joined the National College of Ireland to start the Cloud Competency Centre in 2012, directing NCI's cloud and data analytics infrastructure, postgraduate programmes, and research & innovation initiatives. Horacio started his career as a true dot-commer working in HPC systems engineering and product marketing for innovation-driven companies Silicon Graphics and Sun Microsystems. He later did a PhD and a postdoc in computer science at the University of Edinburgh. He is part of the management committee of the CERCIRAS COST Action.

Outline

Algorithmic skeletons systematically abstract commonly-used structures of parallel computation, communication, and interaction. By seamlessly employing skeletons, structured parallel programs ought to be conceived as two separate and complementary entities: computation, which expresses the calculations in a procedural manner, and coordination, which abstracts the interaction and communication.

This lecture series shall discuss the 30+ year evolution of structured parallel programming with emphasis on industry and academic programming frameworks and its links to modern parallel architectures. It provides a brief introduction to structured parallel programming using the Message Passing Interface (MPI) collectives. Students will gain hands-on experience in developing efficient parallel applications by harnessing the power of collective operations offered by MPI. Emphasis will be placed on understanding the fundamental concepts, principles, and best practices for utilising MPI collectives effectively in parallel programming.

Lab requirements

Computer with Linux and general software development tools. Students will be given instructions on how to download and install a C code base.

Slot 1 (Lecture)

Overview of parallel computing paradigms; Introduction to distributed memory systems; Motivation for using MPI for parallel programming; Structured Parallelism 101: Parallel Software Patterns, and Skeletons

Slot 2 (Lecture and Laboratory)

Parallelism and MPI 101: Overview of the Message Passing Interface (MPI) standard; MPI communication model and process topologies;

Practical: MPI Hello World and Basic point-to-point communication with MPI_Send and MPI_Recv.

Slot 3 (Lecture)

Understanding collective operations and their significance in structured parallel programming; Classification of MPI collective operations; Overview of MPI collective communication routines

Slot 4 (Lecture and Laboratory)

MapReduce and Map & Reduce; Understanding reduction operations and their purpose; MPI_Reduce function and its variants

Practical: Performing basic reductions (e.g., sum, min, max) using MPI Collectives; Map, reduce, scan, gather.

Recommended preparatory reading

- Marc Snir: Technical perspective: The future of MPI. *Commun. ACM* 61(10): 105 (2018) https://doi.org/10.1145/3264415
- Peter Pacheco, Matthew Malensek. *An Introduction to Parallel Programming*. Morgan Kaufmann; 2nd edition (27 Aug. 2021). ISBN: 0128046058
- Horacio González-Vélez, Mario Leyton: A survey of algorithmic skeleton frameworks: high-level structured parallel programming enablers. *Softw. Pract. Exp*. 40(12): 1135-1160 (2010) https://doi.org/10.1002/spe.1026

Lecture Series 3: Timing Analysis of Parallel Real-Time Systems

Instructors: L. Miguel Pinho, T. Carvalho, ISEP, Porto, PT

- Luis Miguel Pinho is a Professor at the Department of Computer Engineering - School of Engineering of the Polytechnic Institute of Porto, and Senior Researcher at the INESC TEC Associated Laboratory. He promotes and leads activities in, among others, real-time parallel programming models, reliable software, and edge computing. He participated in more than 25 research projects, was project coordinator of the FP7 R&D European Project P SOCRATES (Parallel SOftware framework for time-CRitical mAny-core sysTEmS) and work package leader in the H2020 project AMPERE (A Model-driven development framework for highly Parallel and EneRgy-Efficient computation supporting multi-criteria optimisation). He has published more than 150 papers in international conferences and journals in the area of real-time embedded systems, and has been general/program chair of several international conferences. He is a member of ISO/IEC JTC1/SC22/WG9, and senior member of ACM and IEEE. He was Editor-in-Chief of the Ada User Journal, and is currently Technical Editor of ACM SIGAda Ada Letters.
- Tiago Carvalho is a researcher and invited professor at the School of Engineering of the Polytechnic Institute of Porto, where he works in activities related to real-time parallel programming and timing analysis. He has a PhD in Compilers and a MSc degree in Computer Engineering from the Faculty of Engineering of the University of Porto (FEUP), where he is an invited assistant professor. Tiago has experience in compiler-related topics such as domain-specific languages and compiler optimizations.

Outline

Real-time systems are those systems where the computing system needs to provide results within specific time intervals (deadlines), emanating from application requirements. The development of high-performance real-time applications on parallel processors is a challenging task, as it requires guaranteeing functional correctness as well as application-dependent non-functional requirements. When considering parallel-based applications, it is important to analyse and understand the impact that the mapping and scheduling of computation have on the real-time response of the application. In fact, different strategies for the mapping of the computation to the parallel threads may produce significantly different interference, leading to different timing behaviour.

Tuning the application and system parameters proves to be one of the most fitting solutions. Nevertheless, the design space can be very cumbersome for a developer to verify manually all possible configurations. This course will discuss a methodology to profile, analyse and configure the timing behaviour of high-performance cyber-physical applications and target platforms. The methodology leverages on the possibility to generate a task dependency graph from the parallel computation, using this information to evaluate, through measurements, different mapping configurations, selecting the one that minimizes applications' response time.

Lab requirements

Computer with Linux and general software development tools. Students will be given instructions on how to download and install custom compilers, runtimes and analysis tools.

Slot 1 (Lecture): Introduction to real-time parallel models, Luis Miguel Pinho

The methodology for the design of real-time systems has been challenged due to the increasing processing requirements of modern cyber-physical systems, and the corresponding use of fine-grained parallelism. This lecture will provide a general overview of real-time parallel computing, including the use of parallel programming models in real-time systems, and the vertical stack of mapping and scheduling parallel computation.

Slot 2 (Lecture and Laboratory): Concurrent and parallel models, Luis Miguel Pinho/Tiago Carvalho

In this lecture/laboratory, students will get familiar with different concurrent and parallel models. Focus will be set on the tasking model of OpenMP, and on how this model can be used to develop parallel programming amenable to real-time analysis.

Slot 3 (Lecture and Laboratory): Methodology and toolset, Luis Miguel Pinho/Tiago Carvalho

In this lecture/laboratory, we will introduce the methodology and toolset to profile, analyse and configure the timing behaviour of cyber-physical systems. The toolset, from the AMPERE project, includes a custom tailored LLVM compiler, a set of static mapping algorithms, and a profiling tool using performance counters.

Slot 4 (Laboratory): Profiling and configuring, Tiago Carvalho

In this laboratory, the students will apply the methodology and toolset in a set of benchmark kernels, related to real-world applications. The laboratory will be based on a generic PC, but a showcase will be presented on the use of the toolset in an embedded platform.

Lecture series 4: Digital Transformation & Quintuple Helix in the Hospitality Industry - A Case Study Approach

Instructor: Erjon Curraj, Digital Transformation Expert, AL

- Erjon is a digital transformation and innovation expert with over 15 years of experience covering academia, the public sector, private sector, and civil society (quadruple helix). He has a Ph.D. with a focus on the effects of digitalization on the performance of SMEs. Presently, Erjon is working as an UNDP consultant, leading the digital transformation of public services, and other digital inclusive projects.
- In the past, he has held prominent roles in ICT project management, which included being IT Director for large companies including Retail/Wholesale, Service, FInancial and Private Academia sector. Erjon also shares his expertise as a lecturer at several universities, teaching subjects like IT Project Management, Intelligent Entrepreneurship, IoT/Automation, and Digital Transformation.

Outline

This lecture series focuses on the critical intersection of digital transformation and industry-academia collaboration within the hospitality industry, using a case study approach to provide practical insights. Upon completion of this lecture series, participants will have a robust understanding of digital transformation within the hospitality industry, the potential role of AI, and the value of industry-academia collaboration. They'll be equipped with knowledge and practical experiences to drive digital transformation initiatives in hospitality settings.

Lab requirements

A device (PC/laptop) with internet access is required. Instructions for downloading and installing necessary AI tools and platforms will be provided before the sessions, focusing on open-source, free, or trial versions of relevant software.

Slot 1 (Lecture and Laboratory): Understanding Digital Transformation and Industry-Academia Collaboration in the context of Hospitality

Theory: Initiate with a comprehensive discussion on Digital Transformation focusing on its application in the Hospitality industry, and how academia and industry can work together in this space. We'll also explore a case study that illustrates the effective application of digital technologies in a real-world hospitality setting.

Practicals: Students will engage in small-group discussions to dissect the presented case study, examining the impact of Digital Transformation within the Hospitality industry.

Slot 2 (Lecture and Laboratory): Introduction to AI and Its Role in Industry-Academia Collaboration, Erjon Curraj

Theory: Delve into the role of AI and Machine Learning in enhancing the industry-academia collaboration within the hospitality sector. The lecture will cover specific AI applications in the hospitality industry, like predictive analytics for demand management and personalized customer experiences.

Practicals: Students will participate in a hands-on session with basic AI tools and platforms to analyze customer review data for a restaurant chain. They will use Python and Jupyter notebook for this exercise.

Slot 3 (Lecture and Laboratory): Technology Transfer and Innovation Acceleration in Hospitality, Erjon Curraj

Theory: Discuss the importance of technology transfer and innovation acceleration in the hospitality industry, with an emphasis on how academia can support industry growth through research and development. We'll examine a case study detailing a successful technology transfer initiative within the hospitality sector.

Practicals: Students will simulate technology transfer scenarios in the context of the hospitality industry, brainstorming potential applications of an innovative technology.

Slot 4 (Lecture and Laboratory): Fostering Collaboration and Knowledge Transfer in Hospitality, Erjon Curraj

Theory: Explore the concept of the Quintuple Helix Model of Innovation and its relevance to the digital transformation of the hospitality industry. We'll discuss how AI and ML can facilitate collaboration and knowledge transfer, backed by case studies from the hospitality sector.

Practicals: Students will use collaboration tools and AI platforms to propose solutions for a given scenario in the hospitality industry, focusing on how multiple stakeholders can collaborate effectively.