Nordic-RSE conference 2024

Thursday, 30 May 2024 - Friday, 31 May 2024

Aalto University Campus

Book of Abstracts

Contents

1

High Throughput Calculations and High Performance Computing

Author: Rohit Goswami[None]

Corresponding Author: rog32@hi.is

Practical effective usage of HPC resources inevitably ends up in the high throughput regime. No matter how well-optimized a computational chemistry code base may be, most of them (e.g. VASP, GROMACS etc) are geared towards single calculations, while journals require "manifold systems" to prove general results. To this end there has been a steady rise in workflow engines, like AiiDA or PyIron or Luigi or Fireworks. These have a variety of biases based on the original use case. Here, via the usage of MongoDB data-bases and Jobflow with Fireworks, we demonstrate how to perform well versioned (via DVC) high throughput calculations across different computational centers. Part of this is also involves generating optimal binaries for different target hardware.

2

Translating MATLAB software to R: my experience with rBAPS

Author: Waldir Leoncio Netto[1]

[1] *University of Oslo*

Corresponding Author: w.l.netto@medisin.uio.no

I'd like to share my 4-year old ongoing effort of translating the BAPS software (Corander, J. and Marttinen, P., 2006) from MATLAB to R. This is a very large undertaking: the complete sofware contains over 41k LOC, the original developers have disbanded and moved on, and the software doesn't even run anymore on modern systems. We have achieved good progress in 400 hours of work over 3 years, but there is still a lot to go before we get to an MVP. I would therefore like to share with the community some of the things I've learned along the way, and hopefully get some feedback on how to speed up development.

3

What every research software engineer should know about floating point and one bit more

Author: Luca Ferranti[None]

Corresponding Author: lucaa.ferranti@gmail.com

Floating point numbers are the backbone and deepest essence of computation. Floating point numbers is how our computer represents and manipulates "real numbers" (wanna know why the quotation mark? come to the talk). Unfortunately, floating point arithmetic is considered too applied by mathematicians and too theoretical by engineers, hence it often neglected in computer science or scientific computing curricula, and considered an obscure and dry field.

That's not true! Floating point arithmetic is a fascinating realm, where all the maths you have learnt is to be questioned again. Floating point arithmetic is not "wrong or approximate math" but its own maths, full of properties and theorems.

This talk is meant to be a dwarf on giants' shoulders, a revisited and colourful Gen-Z edition of the legendary gem "What every computer scientist should know about floating point arithmetic". In this talk, we will open the pandora box and demistify floating point numbers, discussing what they actually are, how they work and why they work. We will also give a few extra bits to get more accuracy when you need it.

After the talk, the audience will know everything they need to know to work with floating-point numbers in everyday tasks. Moreover, after the talk, you will never be able to type 0.1 in your computer without getting chills through your spine, knowing you are scratching the surface of a brave new world.

4

(Failed) approaches for lifting the general coding competence in a diverse research group

Author: Konstantin Stadler[1]

Co-authors: Candy Eugenie Charlotte Anquetil Ep Deck [1]; Ruslan Zhuravchak [1]

[1] *NTNU - Industrial Ecology Programme*

Corresponding Authors: ruslan.zhuravchak@ntnu.no, candy.deck@ntnu.no, konstantin.stadler@ntnu.no

The Industrial Ecology Digital Lab (https://www.iedl.no) is a small RSE group (7 people) within NTNUs Industrial Ecology Programme (https://www.ntnu.edu/indecol - with around 70 people, organized into eight Professor groups). One of the main purposes of the lab is to support the development of novel research code. As part of that support, we aspire to lift the general coding competence of the whole group.

Our efforts run in parallel to more institutionalized training efforts, like Code Refinery and the Carpentries but also specialized courses (such as HPC or specific Industrial Ecology related courses). We want to bring our students and stuff to a base level to make use of these more intensive course, but also take them up from the course and show them how apply and extend their skills. This has proven to be a challenge, as the group is quite diverse in terms of coding skills and research needs, but most critical due to the high turn-over rate within the group.
Thus is is impossible to create some kind of common basis from which you can lift the competence of the whole group. Over the years, we have tried different approaches to address this issue. These range from structured tutorial series, to informal lunch gatherings, to drop-in sessions. All had their advantages and disadvantages, but none really worked out.

We now settled for some user-centered approach, which is based on an "Open Support" system. The initial feedback is quite positive, but we are still in the process of evaluating the new system.

In this discussion session, we want to present our past and current approaches. Most importantly, we want to discuss with the audience what they think is the best way to lift the general coding competence in a diverse group and hopefully gather new ideas to address this issue.

5

Enhancing Research Efficiency: IndEcx a Web Application for Project Storage and Retrieval

Authors: Hazim Hussein[1]; Konstantin Stadler[2]

[1] *Norwegian University of science and technology*

[2] *NTNU - Industrial Ecology Programme*

Corresponding Authors: konstantin.stadler@ntnu.no, hazemhussein@windowslive.com

In the dynamic landscape of research, efficient project storage and seamless retrieval are essential for driving progress and innovation. However, researchers often face challenges in organizing, accessing, and sharing their projects and associated data effectively. To address these challenges, we introduce a novel web application designed to streamline project management and foster collaboration among researchers.

Our web application offers researchers a centralized platform to store, organize, and retrieve their projects within a comprehensive database. Built with user-centric design principles, the application provides an intuitive and easy-to-navigate interface, enabling researchers to effortlessly store project details, including description, collaborators, keywords, and corresponding files. Through advanced search and filter functionalities, users can efficiently explore their own projects and seamlessly navigate through other projects stored within the database.

Key features of our web application include:

- Convenient project storage: Researchers can store and access their projects anytime and anywhere.
- Intuitive interface: The user-friendly interface ensures ease of use, allowing researchers to quickly input, update, and retrieve project information.
- Advanced search capabilities: Researchers can perform targeted searches based on various criteria, such as project title, keywords, collaborators, and project dates.
- Collaboration tools: The application facilitates collaboration among researchers by enabling shared project access.
- Scalabe code: The code is easily scalable and configured to adjust to the different needs of different research teams.

By leveraging the capabilities of our web application, researchers can foster innovation and inspiration through other projects, aid their project management processes, and foster collaboration across teams. This contribution aims to showcase the potential of our innovative tool in advancing research practices and advancing scientific discovery.

6

Ten simple rules for evaluating the security of research software

Author: Enrico Glerean[None]

Co-authors: Radovan Bast ; Richard Darst [1]

[1] *Aalto University*

Corresponding Authors: radovan.bast@uit.no, richard.darst@aalto.fi, enrico.glerean@aalto.fi

Software security is a familiar concept to anyone using digital devices, often highlighted by frequent and annoying prompts for urgent software updates. When it comes to research software, security takes greater significance: researchers are often working on confidential projects and handling sensitive research data such as personal data. When it comes to security, researchers, who often depend on externally developed tools, are faced with a dilemma: turn to their local overworked IT-security teams to thoroughly assess the security of their code and pipelines or hope that the software libraries and tools they are using are not exposing their confidential projects to the public internet, or undermining the integrity of their files and computer systems. In this talk, we propose a set of rules that researchers can use to self assess the security of the research software they use. After outlining a framework for identifying the potential level of risks, we showcase examples and solutions when it is mandatory to make sure that there are no data leaks, no unwanted access by others, no data corruption. We hope that our proposed checklist will grow and adapt, becoming a living document that provides ongoing value to the research software community.

7

Geant4 : a modern toolkit for the simulation of the passage of particles through matter

Author: Tetiana Malykhina[1]

[1] *Aalto University, Finland*

Corresponding Author: tetiana.malykhina@aalto.fi

The study of processes occurring in a matter when ionizing radiation passes through it is important for solving various problems. Examples of such problems are applied and fundamental tasks in physics, chemistry, material science and technology, biology, nuclear medicine and so on.

Computer modeling makes it possible to perform preliminary computational experiments in cases where real experiments are dangerous or expensive, etc.

An overview of some features of the Geant4 toolkit [1] is presented in the report. Examples of code developed for nuclear medicine, detectors development and applied physics are presented. Some tools for creating an application with GUI and data processing are presented in the report.

[1] https://geant4.web.cern.ch/

8

Deploying Open Source LLMs for on site usage

Author: Thomas Pfau[1]

[1] *Aalto University*

Corresponding Author: thomas.pfau@aalto.fi

LLMs have become a tool used by many researchers in a wide variety of tasks and several libraries are available to facilitate access to the most common LLMs. At the same time many workstations used by researchers don't have the capacity to run llms locally and at the same time researchers are hesitant to feed potentially sensitive data to models hosted on external webservices like Azure or OpenAI. We have set up a local llm deployment, based on kubernetes, FastAPI and llama.cpp. The deployment provides several models along with a self-service checkout for researchers to set up their own API keys. While the service, currently is not intended for high throughput inference, it can serve as a testing ground and can easily be extended.

9

A comfortable interactive setup for scientific computing using VIM

Author: Marijn van Vliet[1]

[1] *Aalto University*

Corresponding Author: w.m.vanvliet@gmail.com

When working on data analysis and other scientific programming tasks, having a code editor and interactive console (REPL) side-by-side is a really powerful setup. It is something that MATLAB popularised (for me at least) and after 12 years is still my preferred way of working. However, I don't use MATLAB. I like VIM and Python. And in this demonstration I want to showcase my workflow and a plugin I helped write to have a good integration between editor and interactive console: https://github.com/jupyter-vim/jupyter-vim

10

Event report - Research Software Engineering: Bridging Knowledge Gaps

Author: Samantha Wittke[None]

Corresponding Author: samantha.wittke@csc.fi

The CodeRefinery project has been invited to participate in the Dagstuhl seminar "Research Software Engineering: Bridging Knowledge Gaps" in mid April 2024 (after submission deadline). This submission aims at summarizing the findings of that event and share information relevant to the Nordic-RSE community.
The main part of the talk will be about the sessions on formal and informal education and training of RSEs in state-of-the-art software engineering research and community building. Also the provision of better access for the software engineering research community to the problems and challenges that research software engineers face in practice will be discussed.

11

Why should organizations have an RSE / RSE group?

Authors: Aurélie Vancraeyenest[1]; Samantha Wittke[None]

[1] *CSC - IT center for Science*

Corresponding Authors: samantha.wittke@csc.fi, aurelie.vancraeyenest@csc.fi

This submission aims at providing a space for the community to discuss topics and questions around suggesting the idea of an RSE/ RSE group to employers. Depending on the audience, we will utilize in-person or remote collaboration tools for the discussion and split into smaller groups if necessary. The outcome of the discussion would be a blogpost for the Nordic-RSE website with suggestions on how RSE enthusiastic academics, specialists, developers and the like could start a conversation with their organizations about offering RSE positions or even starting an RSE group.

12

Presenting InfraVis, The National Research Infrastructure For Data Visualization

Authors: Filip Berendt[1]; Mario Romero[1]

Co-authors: Björn Thuresson [1]; Lonni Besançon [2]

[1] *KTH Royal Institute of Technology*

[2] *Linköping University*

Corresponding Authors: lonni.besancon@liu.se, marior@kth.se, thure@kth.se, berendt@kth.se

InfraVis is the national Swedish infrastructure for the visualization and analysis of data from all academic domains. InfraVis supports its users on three levels with varying degrees of depth and effort: from a few hours without user fees to several months with co-funding. In this paper, we document and analyse our approach to support levels to extract lessons learned for the reader.

InfraVis is a human-resource infrastructure. It's distributed across 9 different node universities, in 8 different cities. The leading/hosting node is Chalmers University, and the others are Lund University, Linnaeus University, Gothenburg University, Linköping University, KTH Royal Institute of Technology, Uppsala University, Mid Sweden University, and Umeå University. Its capacity to provide service is based primarily on the know-how of the people who operate the infrastructure. In this respect, we are different from research infrastructures that provide access to rare high-end equipment, such as supercomputers or electron microscopes. At most, the infrastructure provides users with, for example, specialized graphics hardware and motion capture environments. The infrastructure addressed a general lack of visualization expertise and literacy among many scientists and researchers in Sweden. While most scientists know how to produce figures from experimental results, they typically don't know how to create interactive visual representations tailor-made to address their analytical and exploratory tasks or how to avoid misleading their audience with accidental visualization misrepresentations. At best, typical researchers may know of tools that approximate these analysis and exploration requirements, but, even then, InfraVis plays a dissemination role amongst those that are not up to date with the latest tools.

Towards fulfilling these requirements, we identified that there would be different levels of support that researchers may need. We defined three levels of support, namely helpdesk support, mid-level support, and in-depth support. Helpdesk support is meant to be under 10 hours of work (often less) and the infrastructure's application experts are not meant to handle user data. If the application experts need to handle the data, we automatically claim it is at least a mid-level support project. In mid-level support projects, application experts are operating directly on a version of the data and are providing visualization services for up to 100 hours. The in-depth support projects are defined as those that will require more than 100 human hours to provide adequate support. In practice we have placed a cap of 600 hours for in-depth support. If needed, we advise users to split off the work and apply over multiple rounds of in-depth support. Furthermore, users need to co-finance, either with direct funds or in-kind contributions, the in-depth support projects.
Within the infrastructure, the people doing the actual ground work of providing research support to the users are referred to as InfraVis Application Expert (IAE/AE).

Defining the criteria for these different levels of projects:

- A helpdesk support project is one where the users have clean data, a clear idea of what about this data they want to visualize, what tasks they want to perform with the visualization, and a short list of visualization tools that may perform the task effectively. Their issue might be that they lack the best practice for visualization or don't have the time to invest in learning the tools or methods they need unsupervised. Their issue may be resolved by the IAE answering a few questions or by providing them the specific tool set up they require.
- A mid-level project is one where the user has most of the data and it is mostly clean and needs to hand over the data to the application expert for further processing and visualizing. The user may have a vision of what the final visualization should look like or what cognition tasks it is meant to facilitate, but does not know how to or the tools that will create that vision. A well-defined mid-level project will typically have the user engage with the IAE with a clear idea of what they want and how it might be achieved on a high level, but lack the specific know-how to realise it.
- An example of an in-depth project is one where users request extensive design and evaluation expertise in interactive data visualizations. A common user for this level of project is one that simply has no idea how to efficiently read and understand their data, not because of a lack of effort, rather that there is no clear established way of doing so. The AE(s) not only handle the data, but are in charge of uncovering user requirements and tasks, creating visual analytics tools to support them, and evaluating the effectiveness of these tools in supporting the identified tasks. It is not unreasonable that the final output of an in-depth project becomes grounds for publishing a paper, presenting the new visual analytics method or tool for the broader community of the users'domain.

A potential user would typically submit a request in less than 5 minutes through our online contact form or talk directly with an InfraVis representative, to present a project proposal. The process is simple and undaunting by design, encouraging potential users to discover what powerful, tailor-made visualization support can do for them, without needless consideration about the time or resource cost.

13

Python-based deep learning for detecting ditches from elevation data

Authors: Holger Virro[1]; Wai Tik Chan[1]

[1] *Landscape Geoinformatics Lab, University of Tartu*

Corresponding Authors: wai.tik.chan@ut.ee, holger.virro@ut.ee

The Estonian Topographic Database (ETAK) is a collection of geospatial vector data layers managed by the Estonian Land Board. In general, this data is generated by digitizing various man-made (e.g. buildings, roads) and natural (e.g. water bodies) geographic objects from satellite imagery. This process is currently largely manual and very time-consuming, which is why methods for automating at least some of the mapping process should be investigated. In recent years, deep learning (DL) models have been used successfully for detecting geographic features from remote sensing data.

This presentation gives an overview of our on-going project, which focuses on detecting drainage ditches in Estonia. We developed a DL model based on the U-Net architecture in PyTorch, which uses lidar-derived elevation data as input. We applied a transfer learning approach to overcome the lack of local training data by pretraining the model on labels originating from a previous Swedish study (Lidberg et al., 2023), after which we finetuned it on a small number of samples generated for Estonia. We also intend to extend the current DL workflow with geospatial packages like Xarray and TorchGeo, so that the resulting model is more convenient to apply at national scale.

References
Lidberg, W., Paul, S.S., Westphal, F., Richter, K.F., Lavesson, N., Melniks, R., Ivanovs, J., Ciesielski, M., Leinonen, A., Ågren, A.M., 2023. Mapping Drainage Ditches in Forested Landscapes Using Deep Learning and Aerial Laser Scanning. J. Irrig. Drain. Eng. 149, 04022051. https://doi.org/10.1061/JIDEDH.IRENG-9796

14

Organizing and Open Source Confererence

Author: Jarno Rantaharju[1]

[1] *Aalto University*

Corresponding Author: jarno.rantaharju@aalto.fi

How do you organize a conference as publicly as possible, enabling participants to edit the plan as they go?

We organized this conference mainly on a private discussion channel. It is not closed, anyone can join, but you need to know about it. In our previous unconferences, participants have done a lot to make the event happen and they most likely could help with this one.

I will talk about using standard open source development methods to build as much as possible of the next Nordic RSE conference and demonstrate what I have set up already. This will be a demonstration and a discussion.

15

How I write Rust and make it look like Python

Author: Radovan Bast[None]

Corresponding Author: radovan.bast@uit.no

In this demo I will show how I combine the type safety and performance of the Rust programming language with the convenience of Python.

This approach can be used to speed up Python code (by porting the bottleneck) or to interface a Rust project with the Python ecosystem.

I will show how to build and package a Rust project so that it can be imported into a Python project without any extra dependencies for the user of the package.

16

Lessons learned from running a local RSE group

Author: Radovan Bast[None]

Corresponding Author: radovan.bast@uit.no

The RSE group at UiT The Arctic University of Norway is now two years old.
In these two years we have tried a number of approaches for outreach, support, and project management and tracking. We had success stories, but also experienced "growing pains".

I propose to share our lessons learned from supporting projects across many academic disciplines. What do we know now that we wish we knew earlier? Let's share and discuss.

This can either be in a short presentation or as part of a discussion.

17

Configure your workstation with Ansible

Author: Taha Ahmed[None]

Corresponding Author: taha@chepec.se

This presentation will demonstrate an Ansible playbook that fulfils many day-to-day computing needs of a researcher, and which can be easily extended. This conference will mark the first public release of this playbook, which I developed and used during the course of my own PhD.
Starting from a server image of Ubuntu 22.04 LTS (Jammy) the playbook installs and configures the i3 tiling window manager, the rofi launcher, lots of software packages and various productivity-boosting customizations. Most of its components (so-called "roles") can be easily adapted to other Debian-based desktop environments.
This playbook is an example of small-scale infrastructure as code (IaC), or should perhaps more aptly be termed configuration as code (CaC), and is meant to be managed by the researcher himself or herself, who should always have the freedom to run software of their choice configured as they prefer.