Nordic-RSE Online Get-Together

Monday 30 November 2020 - Wednesday 2 December 2020

Book of Abstracts

Contents

Lightning talks session / 7

RSE work done at the Oslo Centre for Biostatistics and Epidemiology

Author: Waldir Leoncio Netto[1]

[1] *University of Oslo*

A short presentation of OCBE, its members, research topics and software produced.

Talks/workshops/events (TBA) / 8

Crash course: Model-based tester ”Modbat”

Author: Cyrille Artho[1]

[1] *KTH Royal Institute of Technology*

Model-based testing automates the generation of test sequences that share similar characteristics (for example, tests adding or removing data to a list).

Modbat is specialized for testing the application programming interface (API) of software. The model used by Modbat is compatible with Java bytecode. The user defines and compiles a model, which is then explored by Modbat and executed against the system under test. Failed test runs are reported as an error trace.

In this one-hour tutorial, we will first introduce model-based testing and explain how Modbat works, and then proceed with a hands-on tutorial as found on the github wiki of Modbat:

https://github.com/cyrille-artho/modbat/tree/master/src/test/scala/modbat/tutorial

Lightning talks session / 9

Aalto RSE group

Authors: Jarno Rantaharju[1]; Richard Darst[2]; Marijn van Vliet[2]

[1] *Helsinki University*

[2] *Aalto University*

Lighting talk introducing the Aalto RSE team and how the team operates.

Talks/workshops/events (TBA) / 10

PRACE Best Practice Guide - Modern-Processors

Author: Ole Saastad[1]

[1] *University of Oslo*

The just released PRACE - Best Practice Guide for Modern processors (ARM Kunpeng & THX2, Intel Skylake and AMD Rome) is just released.
A short introduction to the guide will be given. Topics cover architecture, programming environment, tuning, performance libraries, performance and introduction to European systems using these processors. A couple of hands on examples and tricks to some of the tools in an optimal way will also be presented.

Talks/workshops/events (TBA) / 11

Ericsson Research and Distributed Applications in the Cloud

Author: Andrew Williams[1]

[1] *Ericsson*

Ericsson has a long history in the telecommunications industry dating back more than a hundred years, but with traditional network infrastructure becoming increasingly virtualized and software defined, and the rise of the cloud, both the way we work and the skills we require is rapidly changing. This change is being rapidly driven by the need to collaboratively develop code that can be meaningfully shared with stakeholders. This talk illustrates the inspiration that *Ericsson Research Cloud Systems and Platforms* (CSP) is taking from the Research Software Engineering community and provides an introduction to one concept for a distributed application runtime that we are working on.

12

What academic bioinformatics could learn from startups?

Author: Dmitrii Borisevich[1]

[1] *Copenhagen University*

Academic biology strives to perform the best research possible. Thirty years ago, it created a foundation for modern-day bioinformatics. But today academic biology suffers a reproducibility crisis. Letters and papers are regularly published in high-impact journals about reproducibility crisis,… and nothing changes. Poor scientific software is considered one of the major causes of the crisis.

From a startup perspective, any academic bioinformatics environments look outdated and generally wrong. CI/CD, shared codebase, code review, Agile, and orientation to the product are seen as necessary to just survive in the startup world. At the same time, these concepts are completely unheard of or even opposed in most of the academic places. Why is it so and what can we do about it? Do we *really* want reproducible research, or do we only want to grumble about it?

The discussion will:

- Start with discussing the experiences of the participants,
- Analyze a trade-off between the benefits and the costs of reproducibility, and how it affects research,
- Compare the benefits of teamwork with the academic “single researcher” mentality, and check how it affects RSE’s outcome,
- Discuss infrastructure and management problems,
- Summarize potential solutions.

Talks/workshops/events (TBA) / 13

R<->python transpiler

Author: Dmitrii Borisevich[1]

[1] *Copenhagen University*

Python and **R** are two major programming languages used for research software development and data analysis in bioinformatics. It is not a symbiotic relationship, but a cold war between the fans of both.

Different tools are available to use R in python and vice versa, but they demand learning *both* languages. This is not easy, and thus rarely adopted.

This lightning talk will pitch an idea of using AST to build a *transpiler* between two languages and showcase a simple demo of converting code written in one language directly into another.

Talks/workshops/events (TBA) / 14

What I learned from publishing my first R package and why you should write one, too

Author: Tobias Busch[1]

[1] *University of Oslo*

Writing an R package has helped me leave my comfort zone and level up my R programming skills. The code I write as a researcher is mostly single-user and single-use. Writing and publishing code meant for others has helped me break old habits and acquire useful new software engineering skills. R has a streamlined ecosystem for package development that supports understanding and adhering to best practices. I will talk about the things I have learned while writing my first R package, why I think writing a package should be a rite of passage for any aspiring research software engineer, and why R is a great tool for this.

Talks/workshops/events (TBA) / 15

MXAimbot - AI-based sample centering for macromolecular crystallography

Author: Isak Lindhé[1]

[1] *MAX IV Laboratory*

What is this?

MXAimbot is a neural network based tool currently in development, designed to relieve researchers of the task of manually and individually centering their samples in synchrotron beamlines for macromolecular crystallography.

How does it do that?

It is a pretty simple CNN trained on a few thousand images from a camera pointed at the loop which holds the samples. These images are annotated with coordinates, height, and width.

Why?

Because the other two alternatives are

1. Manual centering by humans, which is boring and tedious and consumes researchers valuable time.
2. X-ray centering, which can cause radiation-damage the crystal.

Talks/workshops/events (TBA) / 16

Is it possible to make code quality more important than the number of published papers in academia?

Author: Sunniva Indrehus[1]

[1] *UiO*

The code quality in academia has a bad reputation. A global measure of the quality of a computational-oriented research group is typically based on the number of published papers and not a stable and well organized code. The latter is crucial for the further development of the scientific quality of the group. Is it possible to make code quality more important than the number of published papers in academia?

Interesting questions that are closely related to my title:
- How to construct a sustainable workflow for groups working where someone has a user and someone a developer perspective? Where goes the line between what type of knowledge is expected from the users and the maintainers?
- Who has the responsibility to teach the academic staff about best software practices? Is it the individual doing computational stuff he(r)self?
- How can the studying programs at the universities speed up and keep track of the "standard"developments in the business? For instance: in 2020 everybody doing some kind of development should be aware of version control and testing.

Talks/workshops/events (TBA) / 17

European Environment for Scientific Software Installations (EESSI)

Author: Thomas Röblitz[None]

We will do a presentation of the EESSI (European Environments for Scientific Software Installations) project including a demo of its current pilot software stack.

In a nutshell, EESSI develops an infrastructure/service which will eventually allow you to use the same scientific software stack on any machine (e.g., Raspberry Pi, laptop, server, cluster, cloud, supercomputer) running on various operating systems (Linux, macOS, Windows) and the software stack is built from sources and can thereby be optimised for the CPU/GPU/interconnect at your machine. Even better you don't even have to install (almost) any software package as the stack will be delivered to you via CernVM-FS a proven solution to distribute software in the WLCG (Worldwide LHC Computing Grid).

The current pilot stack can be easily tested via Singularity, supports ARM, Intel and AMD processors and includes scientific software packages such as GROMACS, OpenFOAM, bioconductor, TensorFlow as well as all their dependencies.