Instrument control library and server for detector construction and testing

plus some other stuff

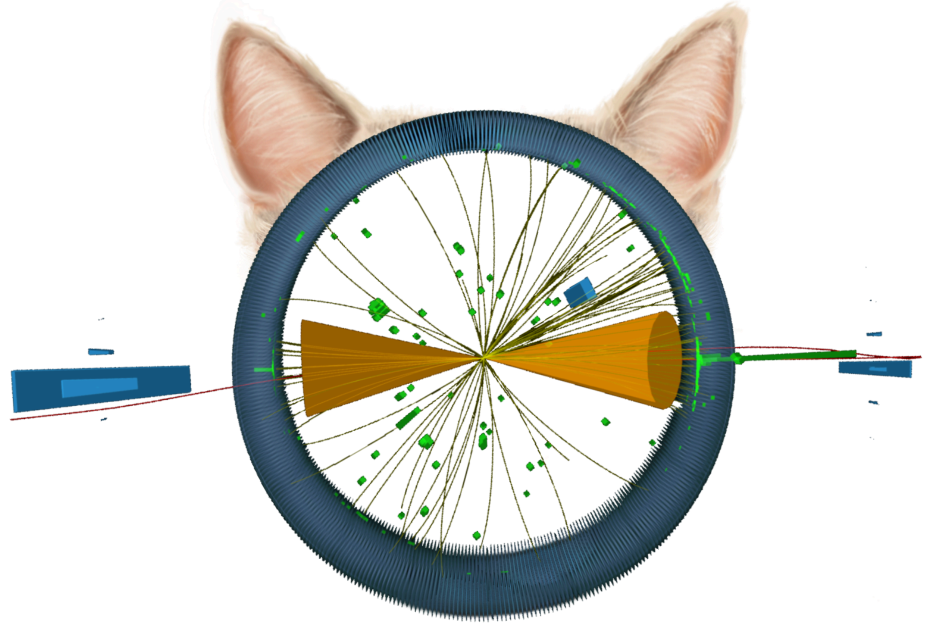
Clemens Lange (Paul Scherrer Institute PSI)

PyHEP.dev

27th July 2023

Intro

- Tenure-Track Scientist in the High-Energy Physics group at Paul Scherrer Institute (PSI) close to Zurich, Switzerland
- Physics analysis interests:
  - PhD performing precision measurements of the tt pair production cross section (ATLAS)
  - Jet substructure techniques and heavy resonance searches
  - Rare and BSM Higgs boson production modes
- Other interests:
  - Pixel detector operation and construction (currently Phase-2 upgrade Inner Tracker modules group convener)
  - Analysis reusability, software containers, and cloud computing
  - Physics analysis tools and training (currently Common Analysis Tools group convener)
  - (by now) sole hepdata_lib developer and maintainer



Overall setup for an 8-channel module test stand

HV power supply/-ies
- 8 channels
- reach -200 V
- ideally also slight forward-biasing possible

LV power supplies
- 8 channels
- reach 10 A each

Chiller
- cool fluid down to 10-15 °C

Control everything through a PC

Coldbox
- Cool down to -35 °C
- Keep modules dry

DAQ setup
- 2x FC7
- 4x FMC
- Imperial Board or µTCA crate
- Ethernet interface

USB
USB
USB
PC
USB
USB

Overall setup example: PSI

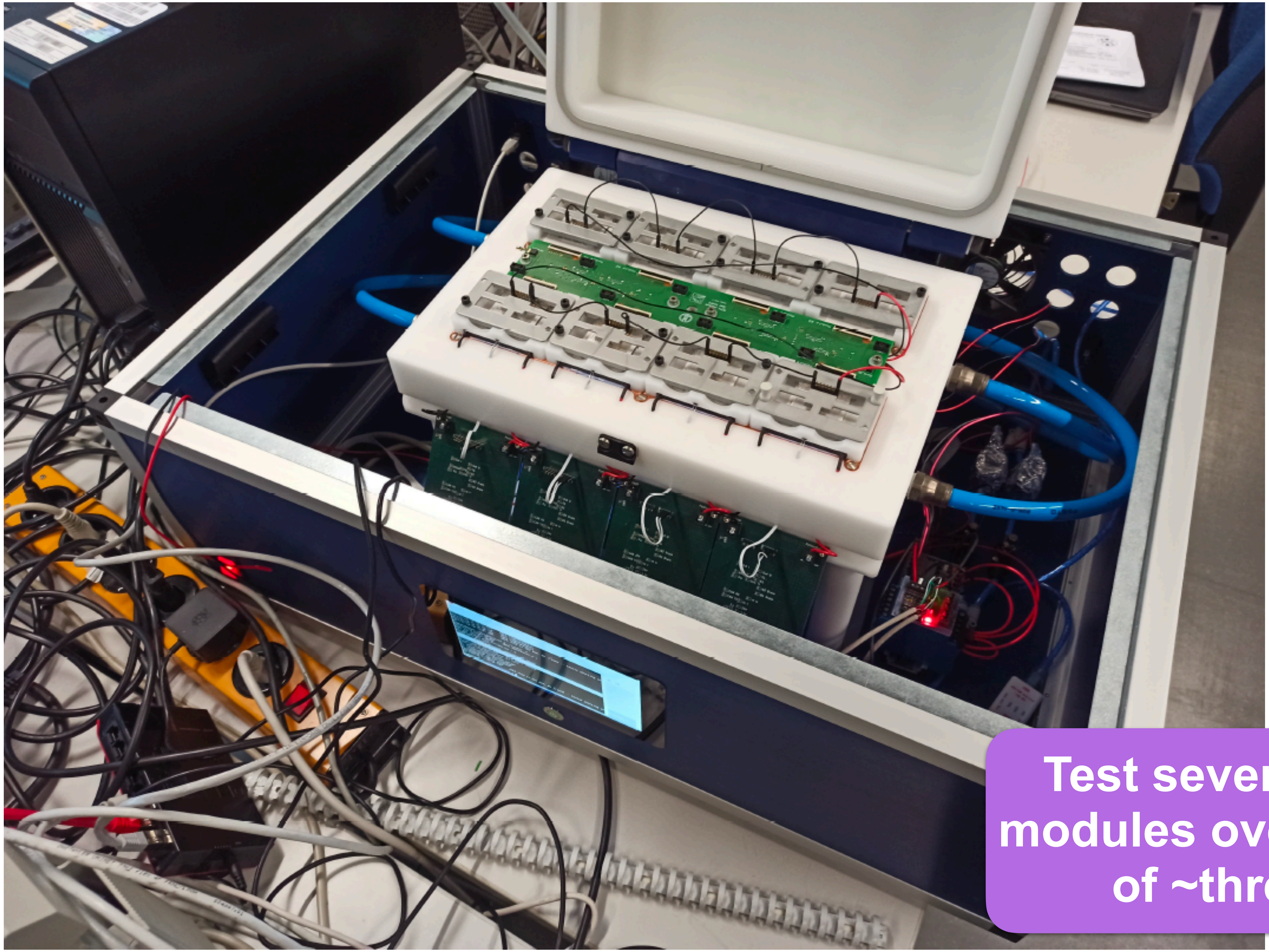
CAEN DT8032 HV (8 channels, fixed polarity)

Lauda Variocool VC 2000

Control everything through a PC

R&S®HMP4040 four-channel instrument
x2
USB
USB
USB
USB
USB
Supported
Supported
Software interface being implemented
Being implemented
CAN BUS
PC
Coldbox
DP merger
Imperial Power Board
FC7
KSU FMC card
Micro SD card
Nano crate
ETH SFP adapter
Power IN
x2

Coldbox



Test several hundred modules over the course of ~three years

Instrument control libraries

- Communicating with hardware can be painful
  - USB communication can be flaky
  - Devices can block if wrong commands are sent
  - Manuals are often wrong and incomplete
  - Difficult to test without actual devices
- Requirements:
  - Robust retry (e.g. redo) and blocking mechanisms (e.g. flock?)
  - Logging and monitoring (interface with InfluxDB and/or Grafana)
  - GUI accessible remotely preventing parallel access → control server
- Several libraries exist, but none seem to have all required features
  - e.g. pymeasure, labRemote, Icicle, Powder, …
  - … so everyone writes their own library (and some even write documentation)

A well-designed library (with typing etc.) would make a difference

Other reasons why I’m here

- Physics analysis requires access to lots of metadata information
  - Require small tools to provide them (web services, utility libraries, …) → this kind of work seems underrated
- Most collaborators will not contribute any code or documentation
  - Even if making/proposing changes is easy, people will not do it → how can we change this?
- A large number of physicists don’t know about Python virtual environments and experiment software makes this more difficult
  - Personal experience: put everything into a container image, deploy as unpacked image, and hide that users are running apptainer
- Python packaging, testing, and library maintenance
  - Open source can be hard and tiring, automation (e.g. GitHub actions) helps a lot keeping things up-to-date and maintainable

PAUL SCHERRER INSTITUT

PSI