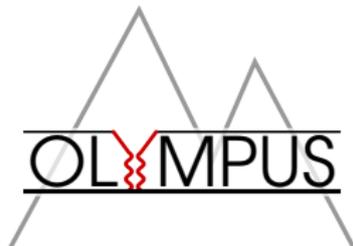


# Offline analysis

Jan C. Bernauer



Collaboration Meeting Januar 2012



Massachusetts Institute of Technology

To get from raw data to a cooked result.

- Raw zebra files are converted to raw root files
  - automatically, with Explora+Plugins (Christian, JCB)
  - Contain all the raw data, but slightly reordered (think “raw steak” instead of “cow”)
  - data structure documented in orawtree.h

To get from raw data to a cooked result.

- Raw zebra files are converted to raw root files
  - automatically, with Explora+Plugins (Christian, JCB)
  - Contain all the raw data, but slightly reordered (think “raw steak” instead of “cow”)
  - data structure documented in orawtree.h
- Each detector group writes Cooker Plugins to convert their part of the raw tree to cooked data.
- Further plugins for tracking, event reconstruction, physics.

To get from raw data to a cooked result.

- Raw zebra files are converted to raw root files
  - automatically, with Explora+Plugins (Christian, JCB)
  - Contain all the raw data, but slightly reordered (think “raw steak” instead of “cow”)
  - data structure documented in orawtree.h
- Each detector group writes Cooker Plugins to convert their part of the raw tree to cooked data.
- Further plugins for tracking, event reconstruction, physics.
- Plugins are run by a cooker program, controlled by recipes.
- Configuration/Calibration data in XML init files. Time stamped.

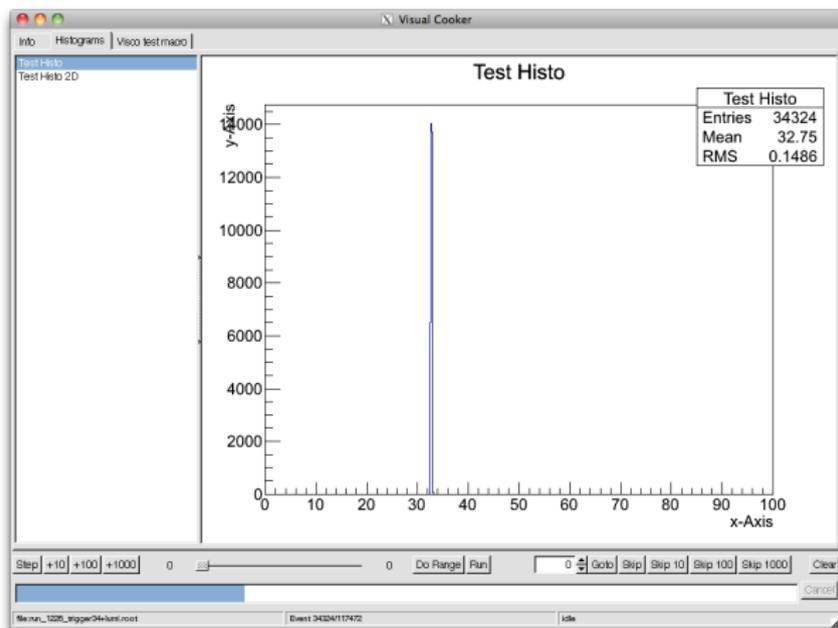
- Command line interface
- For bulk processing

```
[~/src/olympus_compile]
bernauer@silversurfer olympus_compile $ bin/cooker --help
Cooker -- Raw to cooked with a recipe
Syntax: cooker [options] <recipe> <source tree> <target tree>
Allowed options::
-h [ --help ]                print help message
-i [ --init ] arg            override initialization xml file
-s [ --start ] arg (=0)     Number of first event to process (starts with 0)
-n [ --num ] arg            Number of events to process
-r [ --recipe ] arg         recipe
-I [ --input_tree ] arg     input root tree
-o [ --output_tree ] arg    output root tree
-v [ --verbose ] [=arg(=1)] Verbose mode (optionally specify level)
-c [ --call ] arg           Call a plugin's function. Needs argument with
                            format: <plugin>:<function>:<arguments>

[~/src/olympus_compile]
bernauer@silversurfer olympus_compile $
```

# Visual Cooker

- Similar CLI as cooker
- Builds on root GUI
- Event navigation
- Shows plugin defined histograms
- Plugins can add their GUI elements
- Tuning of plugins
- Event display



- Cooker embedded into Explora
- ONLINE analysis while data is taken!
- Explora window will display your histograms!
- Same code as for Cooker/Visual Cooker. Test with Visual Cooker!
- WE NEED DATA MONITORING FOR ALL COMPONENTS.

## Some Plugins

- Copy: Copies Branches/File Objects from source to target.
- Macro: Allows to prototype Plugins / write simple things as a root macro. Fast turn around!
- SlowControl: Unpacks slow control data.
- SCLumi: Luminosity from gas flow, temperature, dead time and beam current

# Repositories

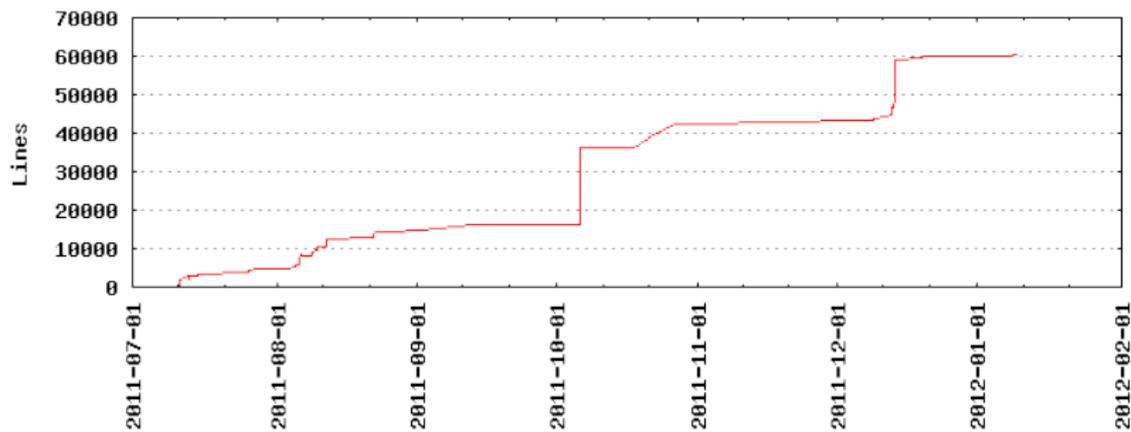
- Git is distributed! This represents only those copies on oweb! There are more.
- By now, everybody should have read access on olympus\_release. If not, contact me.

- Git is distributed! This represents only those copies on oweb! There are more.
- By now, everybody should have read access on olympus\_release. If not, contact me.
- Repositories:
  - olympus\_release: The main one. All have read access. JCB pushes updates
  - olympus\_lumigem: Lumi-GEM work area
  - olympus\_WC: Wire chambers
  - olympus\_MC: Monte carlo
  - olympus\_recon: Track reconstruction
  - olympus\_SM: Symmetric Moeller
  - olympus\_MWPC: MWPCs for Lumi
  - Several personal repos

## All committers (alphabetical)

- Ozgur Ates
- Jan C. Bernauer
- Jürgen Diefenbach
- Christian Funke
- Douglas Hasell
- Axel Schmidt
- Alexander Winnebeck
- + Some from SVN

# Lines of code



- Framework: Usable now, a lot of ideas for extensions.

# Status of code

- Framework: Usable now, a lot of ideas for extensions.
- ToF: Almost feature complete

# Status of code

- Framework: Usable now, a lot of ideas for extensions.
- ToF: Almost feature complete
- LumiGEM: Fast moving, new features every day.  
(Need to pull?)

# Status of code

- Framework: Usable now, a lot of ideas for extensions.
- ToF: Almost feature complete
- LumiGEM: Fast moving, new features every day.  
(Need to pull?)
- WC: basic Hit-finding working. Need data quality monitoring (need to pull?)

- Framework: Usable now, a lot of ideas for extensions.
- ToF: Almost feature complete
- LumiGEM: Fast moving, new features every day.  
(Need to pull?)
- WC: basic Hit-finding working. Need data quality monitoring (need to pull?)
- MWPC: Code exists, but needs to be committed to GIT (Dennis?)

# Status of code

- Framework: Usable now, a lot of ideas for extensions.
- ToF: Almost feature complete
- LumiGEM: Fast moving, new features every day.  
(Need to pull?)
- WC: basic Hit-finding working. Need data quality monitoring (need to pull?)
- MWPC: Code exists, but needs to be committed to GIT (Dennis?)
- SM: No code! Need something until Friday.

# Status of code

- Framework: Usable now, a lot of ideas for extensions.
- ToF: Almost feature complete
- LumiGEM: Fast moving, new features every day. (Need to pull?)
- WC: basic Hit-finding working. Need data quality monitoring (need to pull?)
- MWPC: Code exists, but needs to be committed to GIT (Dennis?)
- SM: No code! Need something until Friday.
- MC: Stand alone works, Plugin coming soon. Better generator etc. needed.

- Framework: Usable now, a lot of ideas for extensions.
- ToF: Almost feature complete
- LumiGEM: Fast moving, new features every day. (Need to pull?)
- WC: basic Hit-finding working. Need data quality monitoring (need to pull?)
- MWPC: Code exists, but needs to be committed to GIT (Dennis?)
- SM: No code! Need something until Friday.
- MC: Stand alone works, Plugin coming soon. Better generator etc. needed.
- RECO: Fast first level, slower second level working, needs testing. Must be integrated. Also: Kalman filter

# Status of code

- Framework: Usable now, a lot of ideas for extensions.
- ToF: Almost feature complete
- LumiGEM: Fast moving, new features every day. (Need to pull?)
- WC: basic Hit-finding working. Need data quality monitoring (need to pull?)
- MWPC: Code exists, but needs to be committed to GIT (Dennis?)
- SM: No code! Need something until Friday.
- MC: Stand alone works, Plugin coming soon. Better generator etc. needed.
- RECO: Fast first level, slower second level working, needs testing. Must be integrated. Also: Kalman filter

## Further things to address

- Cleanup (Everybody, JCB)
- Documentation (EVERYBODY)
- Install helper filers / Sanitize pathes (JCB)
- More CODE!