

# Software Status

MarlinTPC, BBQ and DOOCS

Oliver Schäfer

LCTPC Collaboration Meeting ,DESY, 2020/01/13



# Cleaning up MarlinTPC

- There are still 2000 ... 3000 (formal) compiler warnings in trunk due to “recent” change to C++11 standard → difficult to see important warnings
- Meanwhile, ilcsoft is already adopting C++17 (no compile errors at least!)
- 30% warnings are fixed by, 30% can be fixed by clang code formatter
- 30% are due to bad style and would have been avoided, had people stuck to the ilcsoft coding rules from 2006, which officially went into MarlinTPC rules
- I implemented these coding rules (some for now) as steering file for clang code formatter. Due to YAML syntax can be integrated into Doxygen documentation → Advantage: changes get documented in SVN and would be seen by LCTPC
- Cleaning is necessary to get MarlinTPC to a working state again and adopt new things from ilcsoft (DD4HEP, MarlinMT), new productivity tools like Github
- SVN web interface gone since end 2019, switch to Github is not straight forward, for now use command line and graphical desktop tools

# Cleaning Up Organization

- MarlinTPC mailing list
  - Not very up to date, please help for your group!
  - Separate mailing of registrants to MarlinTPC SVN repository
- Test beam data to the GRID!
  - Documentation on MarlinTPC Wiki still seems up to date
  - Recently updated overview of existing data
  - Available data can be **read** without certificate on the NAF (e.g. for analysis jobs)  
[/pnfs/desy.de/ilc/tpc/...](#)
  - **Full access** with certificate by grid tools at [ilc/tpc/...](#)
  - Should be uploaded as early as possible, reliable storage solution for scientific data
  - At DESY, integrated in data taking for each run, use available fast data connections
  - While no slow control data in DB, ASCII dumps should be put there, too (better than nothing)

# MarlinTPC, BBQ Recent Events

- Most active committer to own branch: Paul
- Amir (Siegen/Bonn) and me (for cleanup) from last year
- Uli? Uwe? Oleksiy? Still things pending?
- Saclay and Japan work on local versions of unknown age for testbeam (likely trunk), no new user accounts
- Help request by Aiko Shoji to overlay consecutive events to from double tracks  
→ writing new processor, lively exchange, so far
- Developing on NAF seems feasible, even from Japan, done for ilcsoft
- Bugfixing request for BBQ event display from ilcsoft in September
  - Turned out to have been fixed by me three years ago, but had not become a tag and thus, was not used by ilcsoft
  - BBQ is still in use for pixels, because of large channel number
  - Also many C++17 warnings

# DOOCS

- Control System Software mostly working, occasional rebuilds
- DOOCS Framework is developing rapidly, new spin-offs (e.g. Chimera-TK)
- MCS4 improved software development and documentation a lot
- Lively conversion towards C++17, new programming methods/tools
- Plans for improvements of our servers are there, some servers & interfaces still need to be written (Conditions Data, SALTRO, Stage), happens whenever there is time
- Maintenance of test beam servers has priority; plans to move to non FLC infrastructure in RZ
- SC data storage has gaps, because of ringbuffer architecture, if feasible switch to LongTerm archivers and get going with conditions data storage, even though it is not used in MarlinTPC, presently.
- SC data was sometimes duplicated as ASCII file by the groups.

# Backup

# MarlinTPC – Vision and Reality

- MarlinTPC should be the place to collect knowledge of the collaboration
- Should be the basis for input to ilcsoft
- Original idea in 2006: same processors for beam tests and ILD
- Standardized workflows help to understand the TPC and the different technologies
- BUT: Currently MarlinTPC is difficult to understand, develop and maintain
- It is currently NOT part of ilcsoft and therefore not used for ILD
- Not much documentation is easily accessible (MarlinTPC Wiki, Doxygen) → improving

# How to clean up MarlinTPC?

- First: I will fix the compiler warnings, hopefully automated by script.
- Then: We should try to collect interesting code from all the old SVN user branches, many have not been used for years
- We can “delete” the other unused branches – they are still available in previous revisions in SVN → it is clearer, what is active work
- Based on the book “Clean code” I will try to rewrite one processor as an example
- C++11 allows to write many structures more compact and clearly
- Changing the directory structure of MarlinTPC again may help, too, because it will be easier to write small, reusable code modules
- If that works well, we can agree within LCTPC on updated coding rules and rework the other code in the same way
- Switch code to use lcggeometry instead of gear



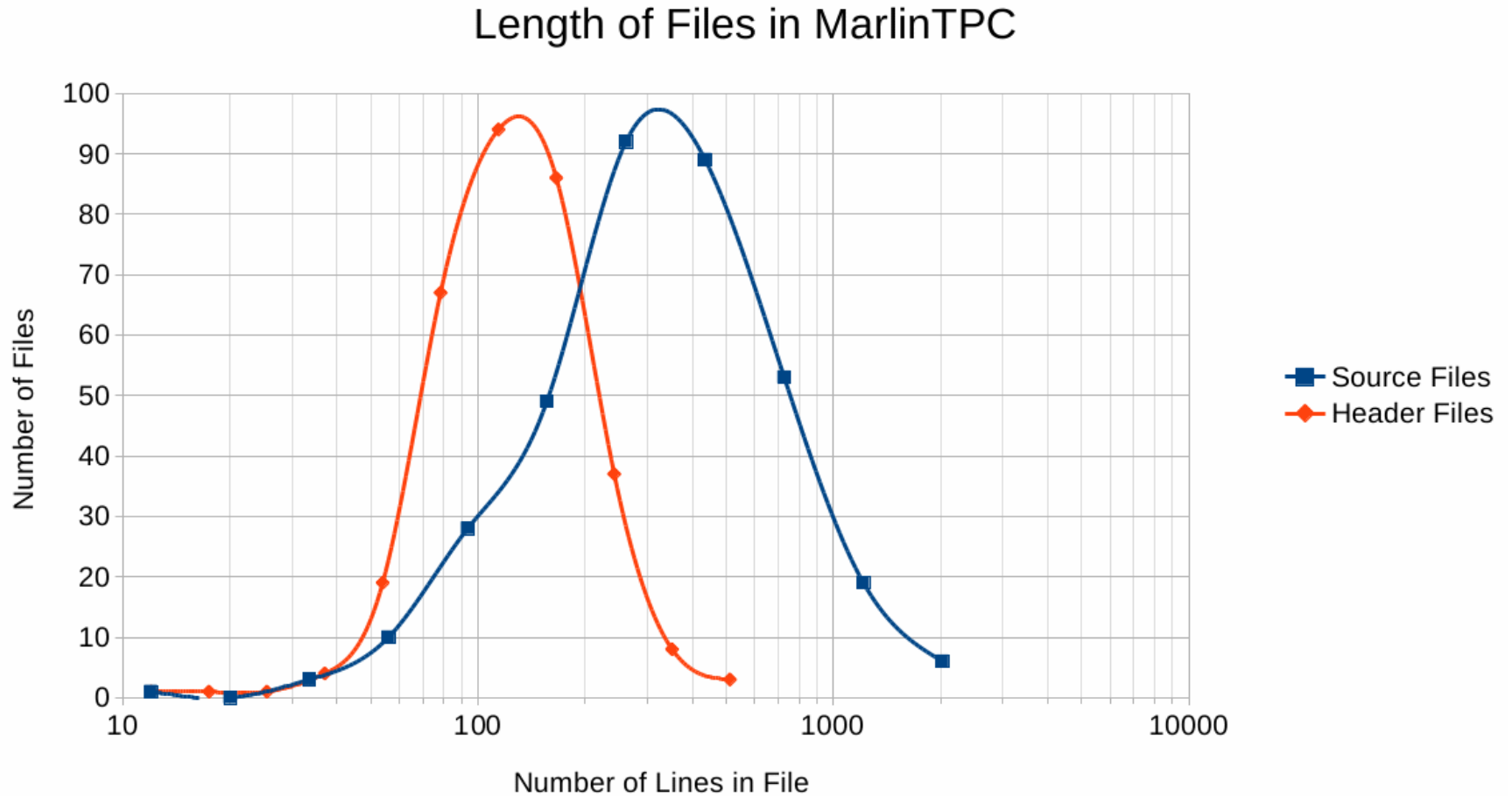
# Documentation

- Is again easily available
- Programming How-To (MarlinTPC Wiki)
- Processor description (Doxygen)
- Work-Flow description (standard steering files)
- LC-Note about standard reconstruction in LCTPC
- Please tell me what you need

# Status of the Code Base

- Number of code files: about 670
- Lines of code, whitespace, and comments: about 150000
- Sounds good and productive, BUT:
- Code base structure were often generated by script with much unused and duplicated code, because nobody dares to delete the unnecessary things
- Many people stuck to the basic processor file layout and produced very long functions (a.k.a. “Spaghetti-Code”)
- Very different coding styles used in parallel

# Status of the Code Base



# Status of the Code Base

## Maximum Line Length in Files MarlinTPC

