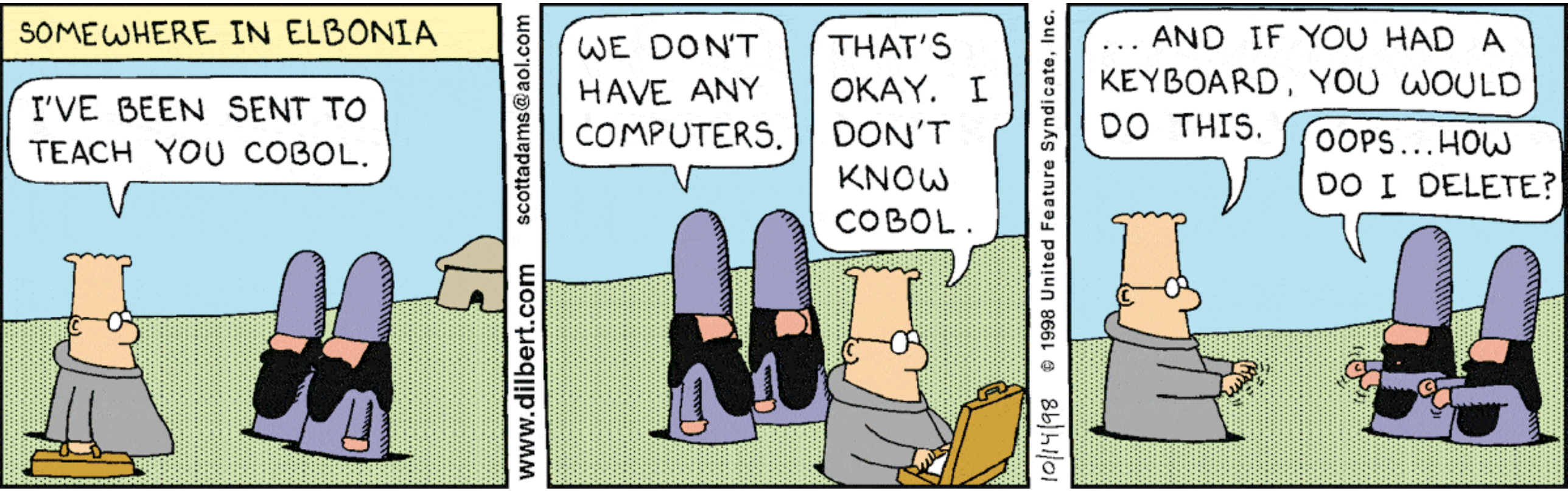
The University Of Sheffield.

THE ROYAL SOCIETY

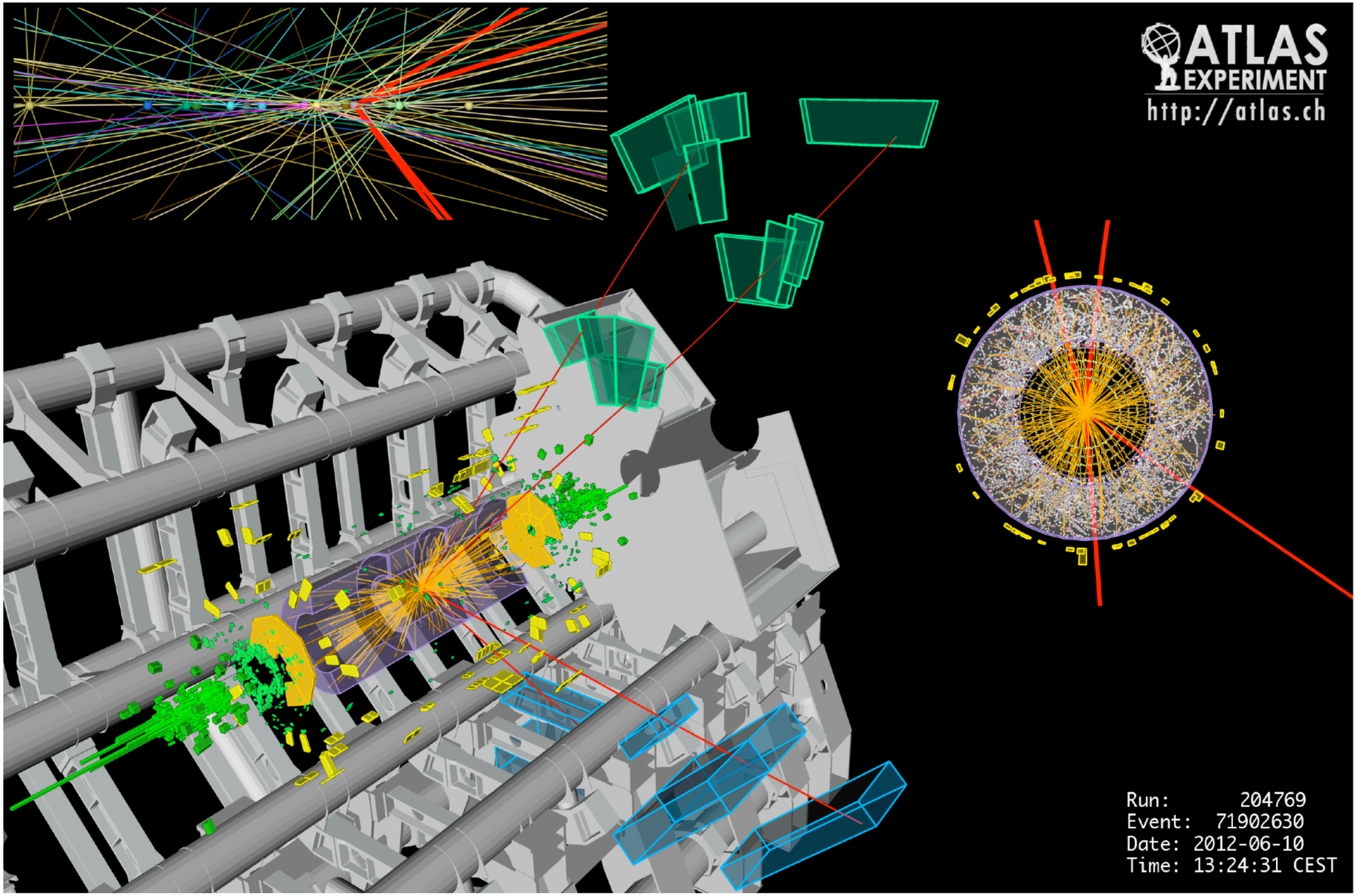
Migrating 4M of C++ to run in multithreaded environment

Christos Anastopoulos
Royal Society University Research Fellow



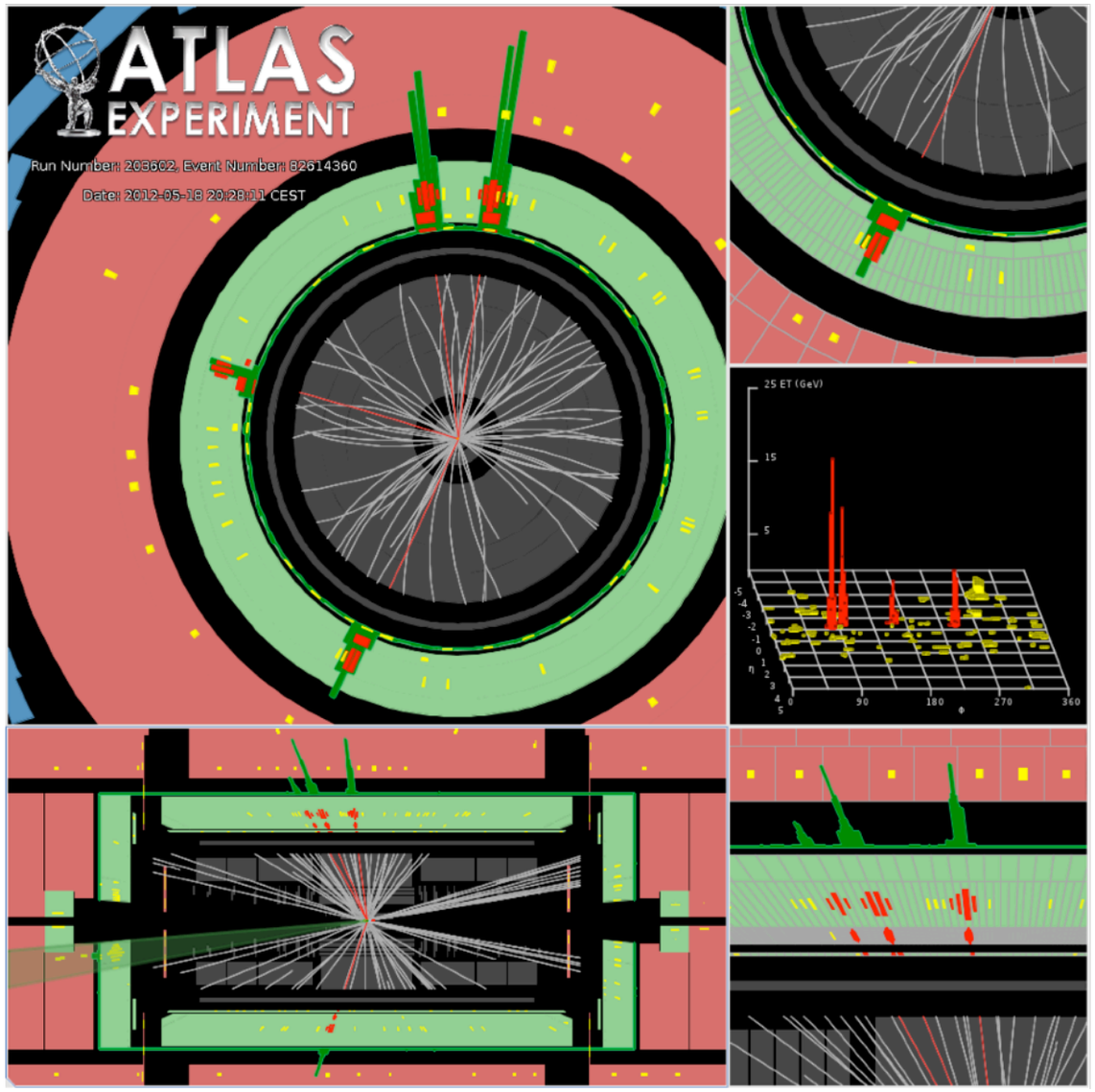
Where these 4M C++ lines are used ?

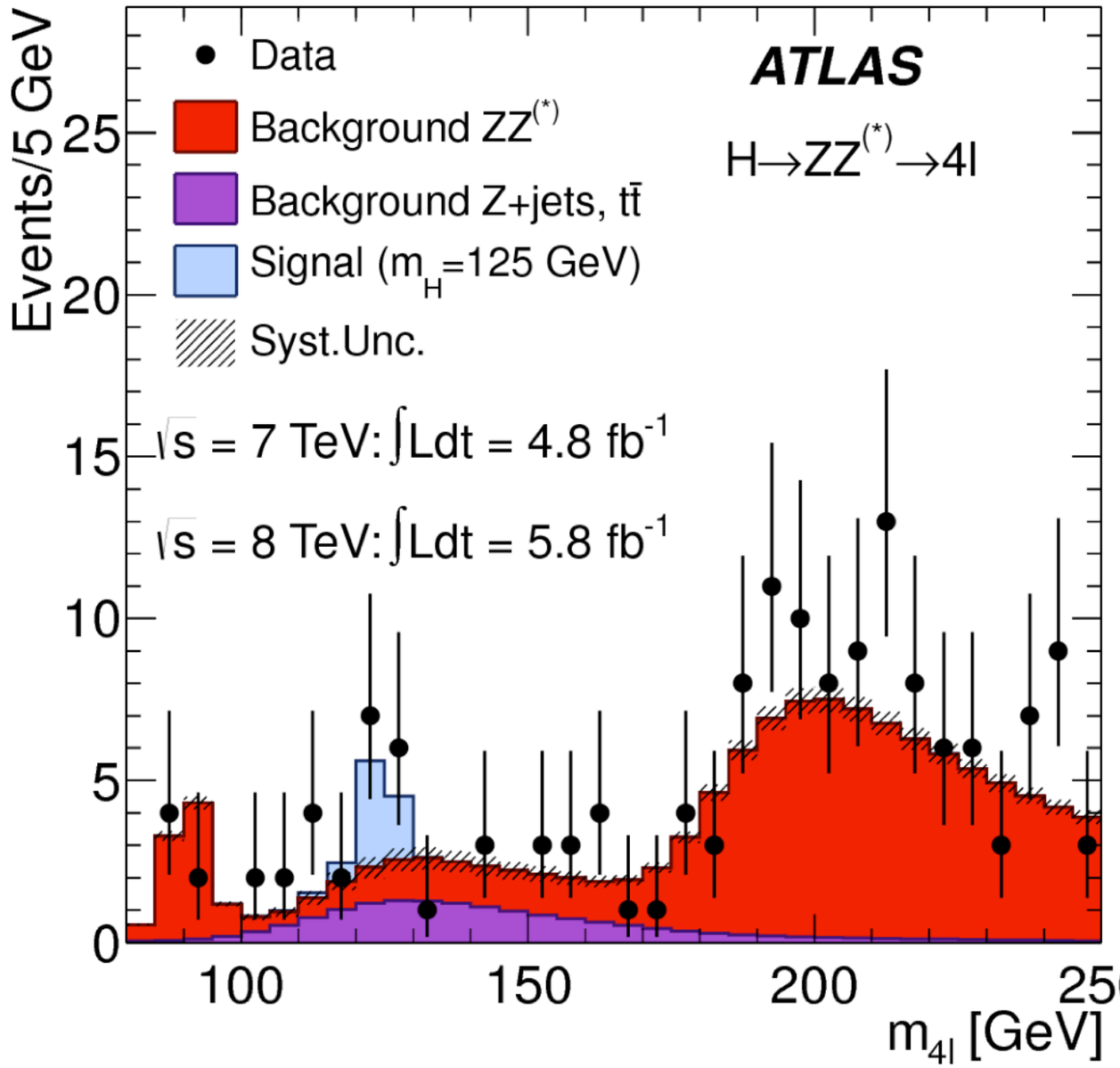
In an experiment like Atlas we reconstruct events.



Where these 4M C++ lines are used ?

Broadly speaking, the purpose of this software is to convert the signals in the ATLAS sub-detectors to "particle candidates".
These then form the input for all ATLAS analyses and papers.



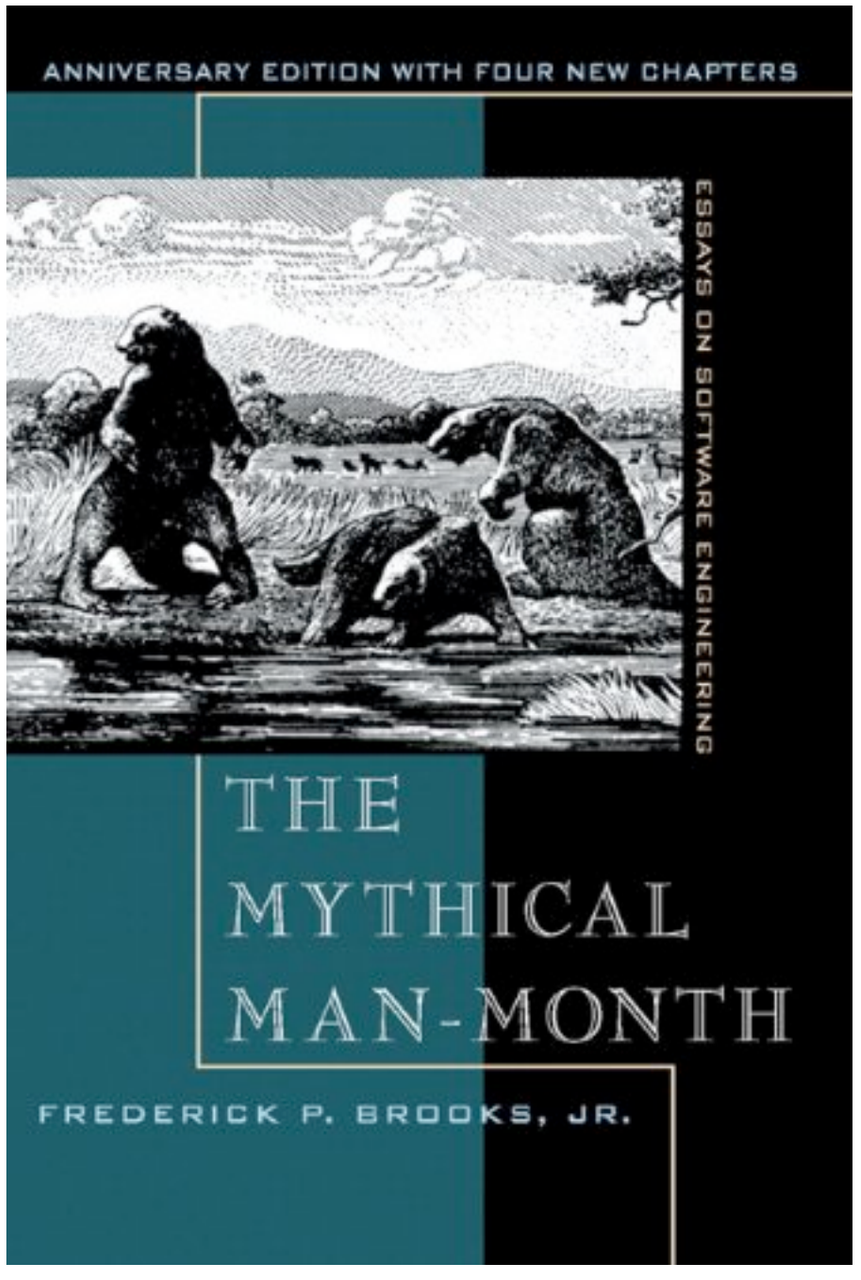


Where these 4M C++ lines are used ?

"Essential" complexity arises from the actual problem we try to solve.
Trying to do non trivial "Physics" with a complex detector

"Accidental" complexity arises from the tools, procedures we use to develop and test our code.

Most of the talk will be more on managing "accidental" complexity



The ATLAS repository

On the 17th December 2018, ATLAS updated the Athena repository to make it public and open-source.

https://gitlab.cern.ch/atlas/athena

atlas › athena

athena
Project ID: 53790

Unstar 161 | Fork 1652

91,777 Commits | **32** Branches | **2,655** Tags | **135.9 MB** Files | **713.7 MB** Storage | **250** Releases

The ATLAS Experiment's main offline software repository

DOI 10.5281/zenodo.2641997 | Doxygen master

LICENSE 109 Bytes — Edit | Web IDE | Lock | Replace | Delete

```
1  The software in this repository is released under the Apache 2.0 license, except where other licenses apply.
```

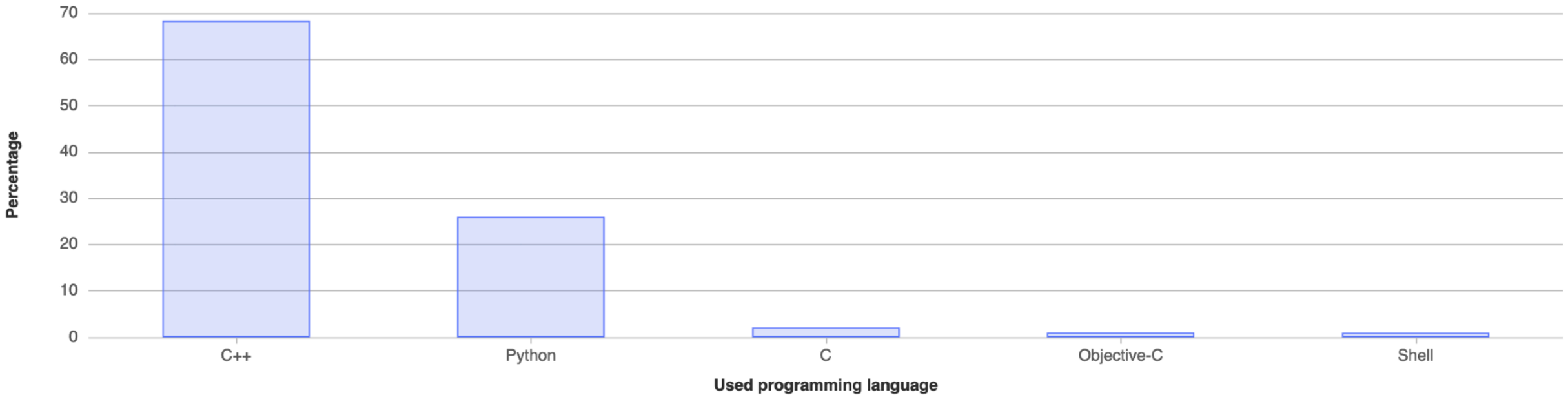
GaussianSumFitter.cxx 54.6 KB — Edit | Web IDE | Lock | Replace | Delete

```
1  /*
2    Copyright (C) 2002-2022 CERN for the benefit of the ATLAS collaboration
3  */
4
```

C++ is the main language
Python is next

Programming languages used in this repository

Measured in bytes of code. Excludes generated and vendored code.



Some further info

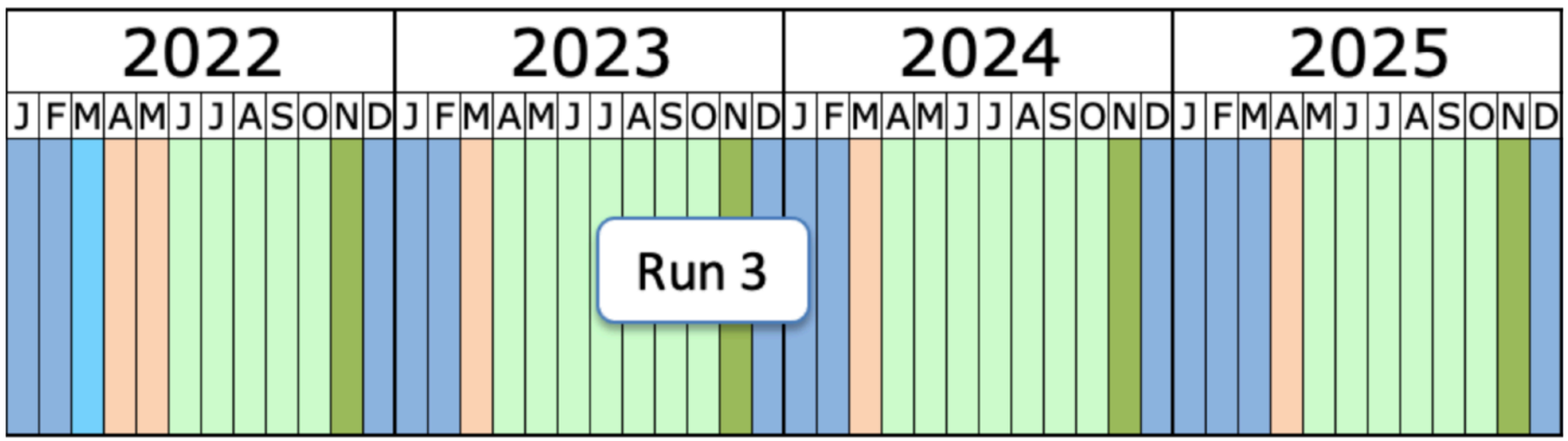
- Main compiler for production gcc, but we also build with clang.
- Main platform x86-64, we started exploring aarch64
- Build System CMake
- Static checkers include flake8 (python), gcc plugins, clang-tidy, cppcheck, coverity (in-progress)
- Leak checkers/profilers : Valgrind, Callgrind, Vtune …
- Issue/feature tracking via JIRA
- C++ style guide : https://atlas-computing.web.cern.ch/atlas-computing/projects/qa/draft_guidelines.html

Time frame

Stable releases used for analysing data, correspond to a particular branch, only bug fixes allowed

The master branch is where we develop new code. We produce "nightly" builds (gcc, clang etc) each day.

The milestone for this talk is the LHC Run-3 data taking period.
Starts in ~ 3 months ….

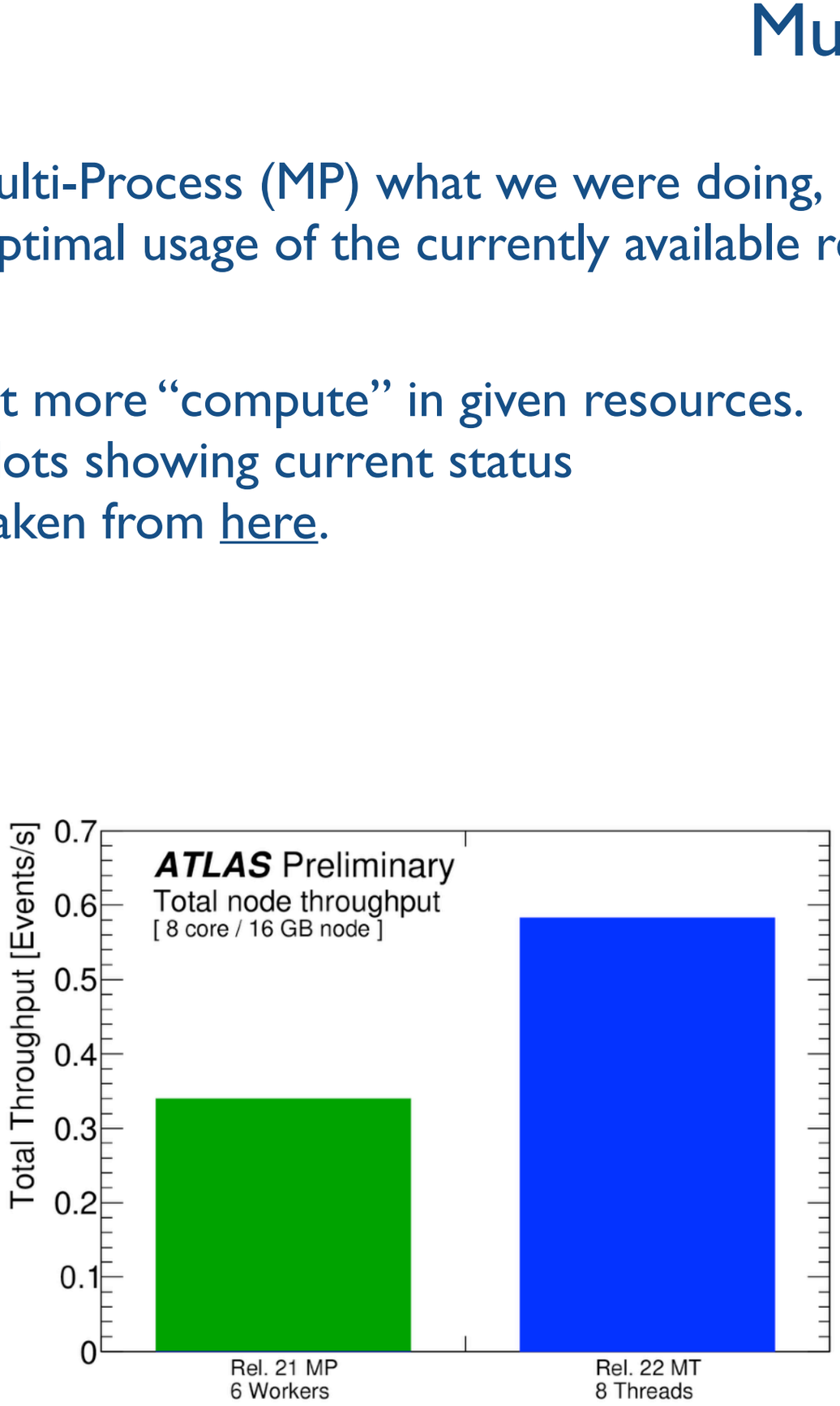


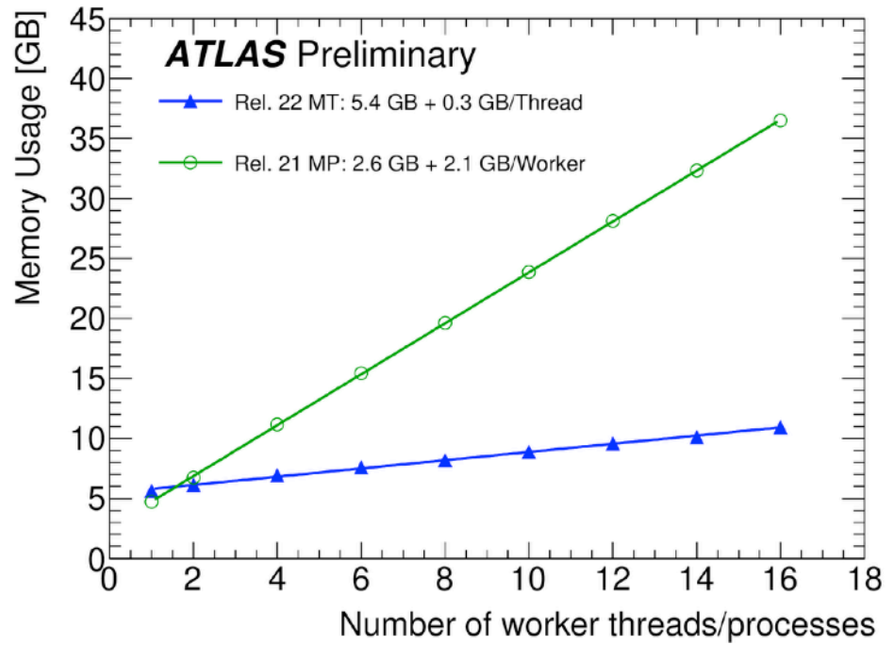
All ATLAS papers to be published in the next few years will use the software discussed.

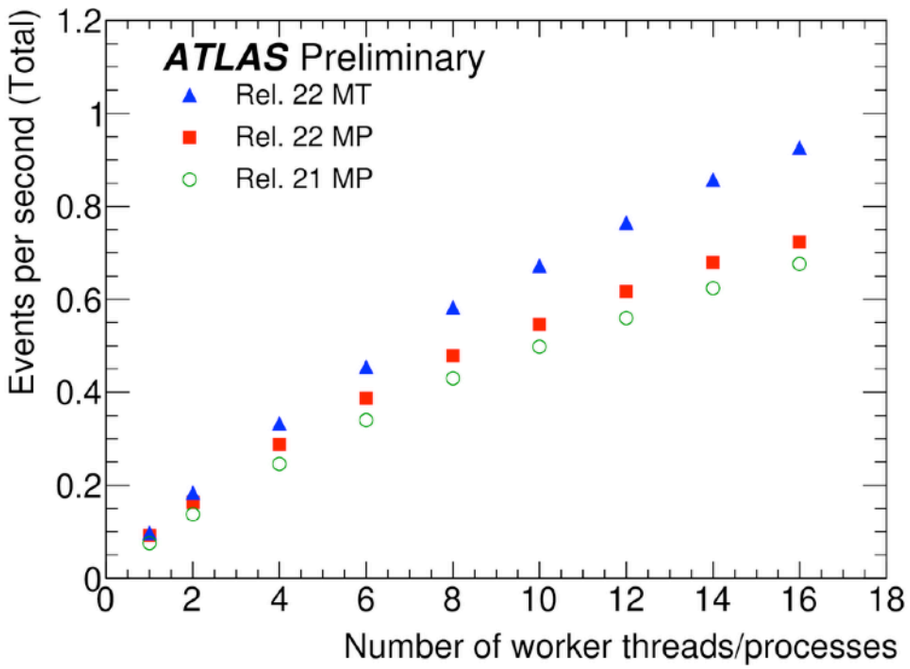
Multi threading

Multi-Process (MP) what we were doing,
Optimal usage of the currently available resources meant we had to migrate to Multi-Threading

Fit more "compute" in given resources.
Plots showing current status
Taken from here.







The problem

Migrate a code base mainly developed with "serial" running in mind
Written mainly pre the C++ "threading" model …

Current estimate is that we touched ~ 1.5 M lines of C++ code …

While at the same time ensuring that the "physics" related output is correct and even improved.

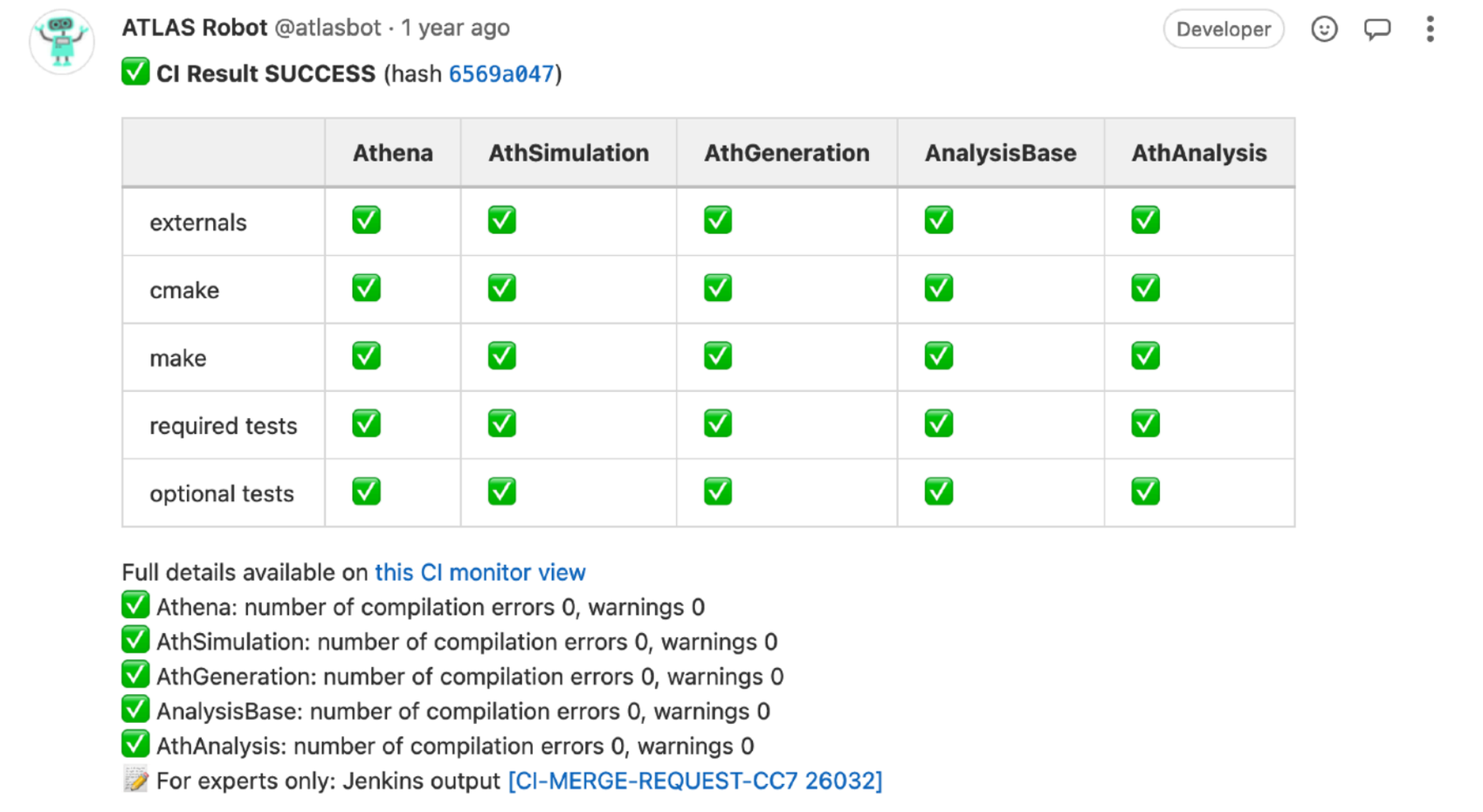
Things that helped

Merge request code reviews

Unit tests : Test the output/behaviour of one module on some “mocked” input.

Integration Tests : Run a few actual events through the full reconstruction chain. Check if number of muons, electrons etc change.

All part of automated “pipeline”. Seems trivial but a huge game changer.



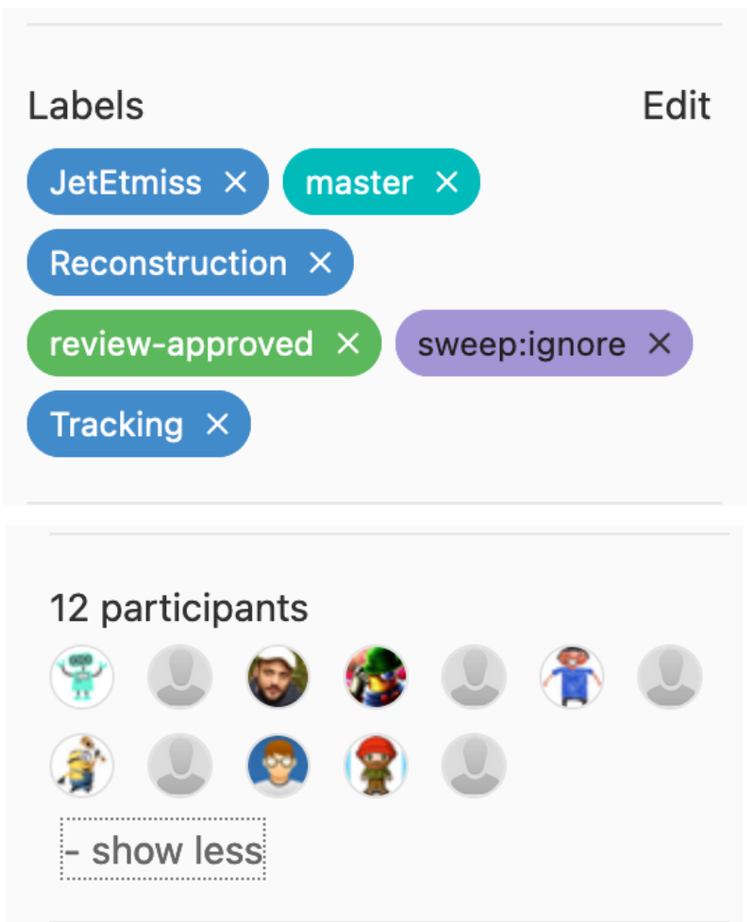
ATLAS Robot @atlasbot · 1 year ago

Developer

✅ **CI Result SUCCESS** (hash `6569a047`)

	Athena	AthSimulation	AthGeneration	AnalysisBase	AthAnalysis
externals	✅	✅	✅	✅	✅
cmake	✅	✅	✅	✅	✅
make	✅	✅	✅	✅	✅
required tests	✅	✅	✅	✅	✅
optional tests	✅	✅	✅	✅	✅

Full details available on this CI monitor view
✅ Athena: number of compilation errors 0, warnings 0
✅ AthSimulation: number of compilation errors 0, warnings 0
✅ AthGeneration: number of compilation errors 0, warnings 0
✅ AnalysisBase: number of compilation errors 0, warnings 0
✅ AthAnalysis: number of compilation errors 0, warnings 0
📝 For experts only: Jenkins output [CI-MERGE-REQUEST-CC7 26032]



Labels Edit

JetEtmiss × master ×

Reconstruction ×

review-approved × sweep:ignore ×

Tracking ×

12 participants

- show less

Things that helped

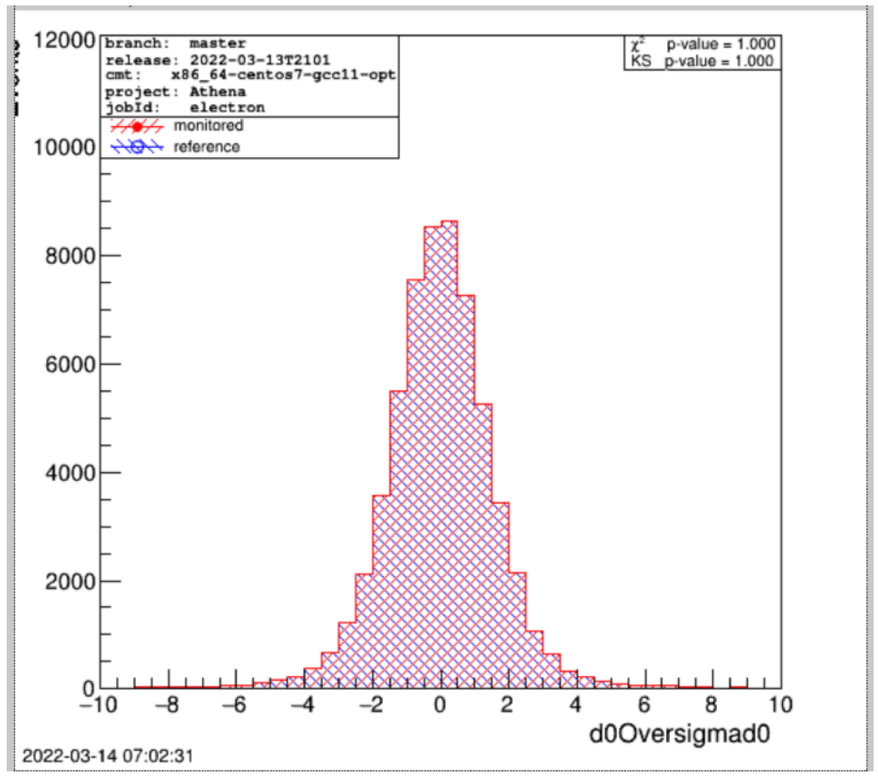
On top of the merge request / Continuous Integration (CI) pipeline.

ATLAS Runtime Testing (ART) : 100s tests running for each “nightly” release (24h turnaround)

Example test output for 1 vs 8 threads:

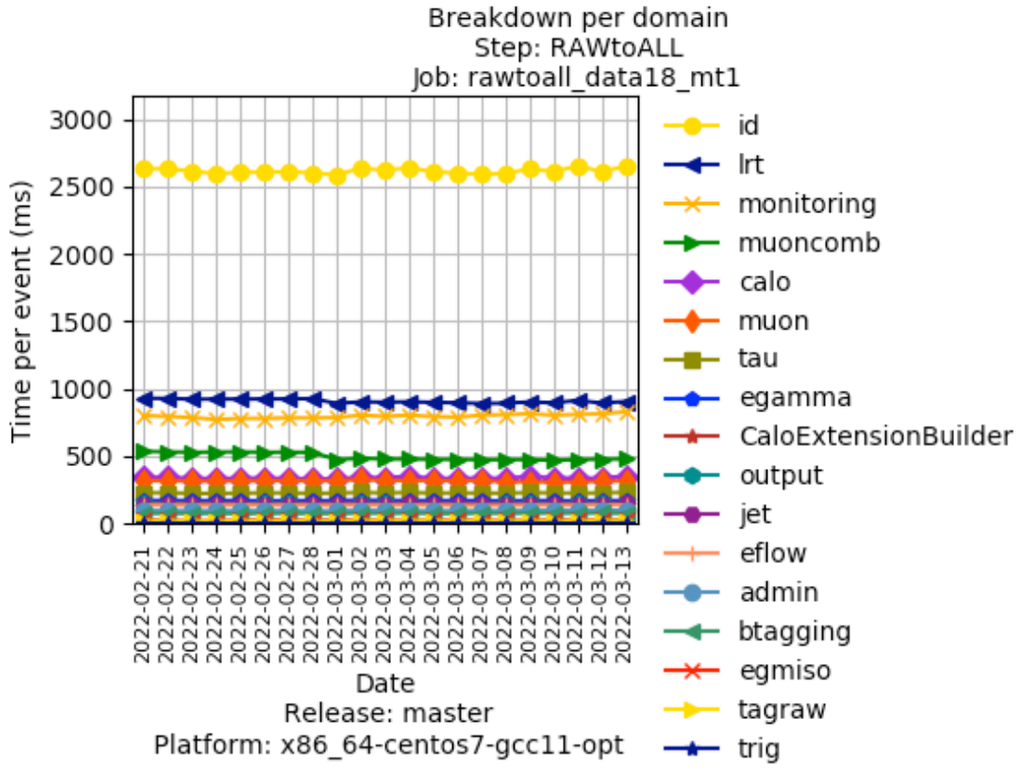
test_recexreco_art_q431_compare1Vs8Threads.sh	1	2	3	succeeded	T2101

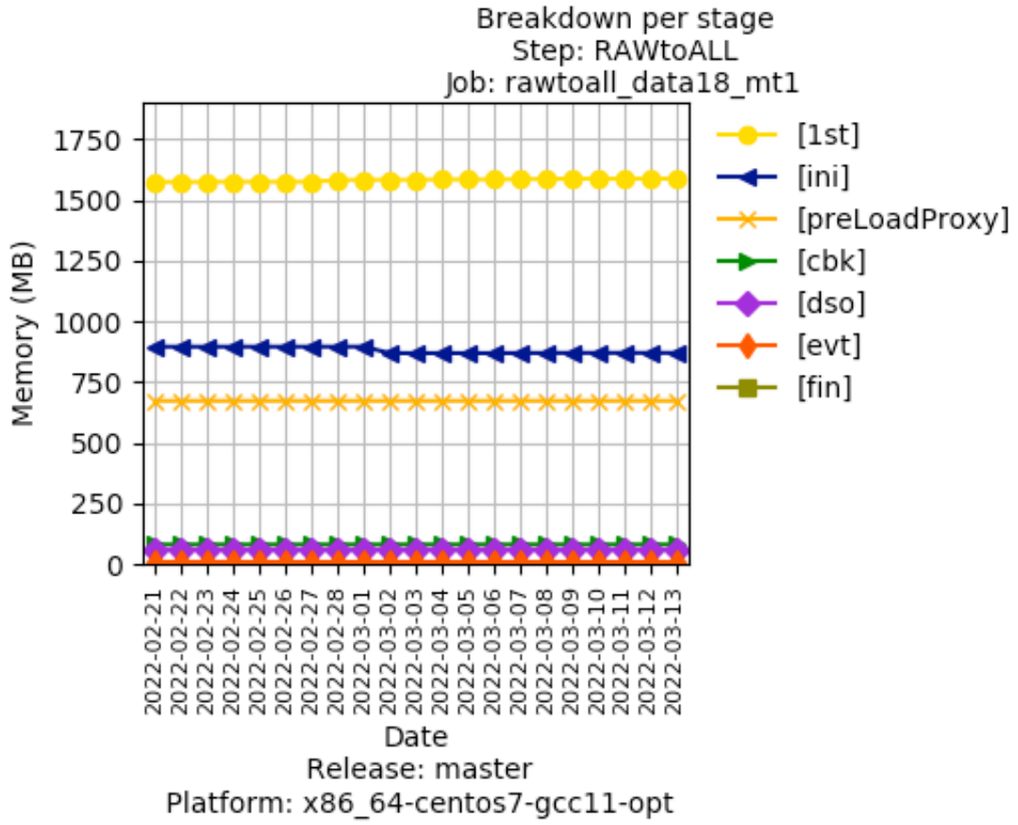
Example test distribution that are used for “physics” using high stats sample.



Things that helped

Constant monitoring or cpu/memory metrics ~ 24 h turnaround





Things that helped

On a weekly basis reconstruct a few million of events from different run periods.
Catches rare issues on taking unexpected paths

- Floating point exceptions
- Dangling pointers
- Rare race conditions.

The final step “Physics” validation, collaboration wide effort where data produced by the software is used in realistic analysis scenarios.

It is not “continuous” but represents “sign off” points during the development cycle.

Things that helped

I am pretty convinced that the effort would have failed without this new machinery we started putting in place ~ 2016.

Code reviews, CI pipeline has been an almost "magic" transformation on the way we develop code.

Coupled with additional testing meant that we had a concrete view of where we are,
the issues we were facing, the effect of any solution.

What follows is a couple of examples from personal experience.

Irreproducible “Muons”

The issue as described in Draft MR (closed) from our software co-ordinator 1 vs 8 threads

```
-bash-4.2$ grep "Muon,  pt" out1.log
#####   Muon,  pt : 2216.89 eta : 1.59942 phi : 0.75371 mass : 105.658 author 4 type : 2 secondary authors:
#####   Muon,  pt : 3214.67 eta : 1.33373 phi : 1.56512 mass : 105.658 author 6 type : 0 secondary authors:  4
-bash-4.2$ grep "Muon,  pt" out8.log
#####   Muon,  pt : 2216.89 eta : 1.59942 phi : 0.75371 mass : 105.658 author 4 type : 2 secondary authors:
#####   Muon,  pt : 3212.1 eta : 1.33373 phi : 1.56515 mass : 105.658 author 6 type : 0 secondary authors:  4
```

In lay terms :
The fitter used for muon trajectories was giving different outputs > 1 threads.
Telltale mark of MT hostile code.

Irreproducible “Muons”

I would probably not have even attempted this without our testing. As I would be effectively touching a critical piece of code “blind”. MR

atlas > athena > Merge requests > !39573

Merged Created 1 year ago by **Christos Anastopoulos** Developer Edit

iPat Utils make it play nice with Athena MT

Overview 9 Commits 6 Pipelines 2 Changes 13 All threads resolved

@rosati , @nkoehler , @wleight is what I kind of had in mind if we want to start understanding the iPat fitter in MT and try to make it behave a bit better, in the sense of the full call chain passing `ATLAS_CHECK_THREAD_SAFETY` etc ...

- Tier0 tests seem to fine (which is a bit of suprise in some sense ...) RunTier0Tests.log
- Make iPatFitterUtils pass the ATLAS thread safety checker
- Update iPatFitter.

The main ideas

- remove `mutable`,
- try to be `const` correct,
- remove "global" state (fitMatrix was acting like a `COMMON BLOCK` in my best description)
- remove "internal" state to a struct `FitProcedure::Cache` following the by now usual trick, at least the one used in `Gx2` and `GSF` to avoid synchronisation.
- In the process also add some more `unique_ptr` usage.
- I had to touch / read a few place so had to also consistently format the code to be able to go through ...

In less than a week we knew the answer.

Edward Moyse @emoyse · 1 year ago Owner

Wow. This really seems to fix a lot of irreproducibilities - well done Christos!

“Trivial” changes to interfaces

Cases where one needs to touch an interface. Many clients. Example MR

athena

atlas › athena › Merge requests › !49663

Merged Created 1 month ago by **Christos Anastopoulos** Developer

IExtrapolator rm legacy / fallback Event Context unaware methods

Overview 11 Commits 3 Pipelines 3 **Changes 93**

Compare master and latest version

93 files +1293 -1351 Expand all files

ATLAS Robot @atlasbot · 1 month ago Developer

CI Result FAILURE (hash `f4e98ab0`)

	Athena	AthSimulation	AthGeneration	AnalysisBase	AthAnalysis	DetCommon
externals	✅	✅	✅	✅	✅	✅
cmake	✅	✅	✅	✅	✅	✅
make	⭕	✅	✅	✅	✅	✅
required tests	✅	✅	✅	✅	✅	✅
optional tests	✅	✅	✅	✅	✅	✅

ATLAS Robot @atlasbot · 1 month ago Developer

CI Result SUCCESS (hash `657c1c25`)

	Athena	AthSimulation	AthGeneration	AnalysisBase	AthAnalysis	DetCommon
externals	✅	✅	✅	✅	✅	✅
cmake	✅	✅	✅	✅	✅	✅
make	✅	✅	✅	✅	✅	✅
required tests	✅	✅	✅	✅	✅	✅
optional tests	✅	✅	✅	✅	✅	✅

CI more or less tells if you forgot something.

The rise of the “robots”?

We consider using more clang-tidy in out work flow.
Example of using modernize-replace-auto-ptr to enforce out “style-guide”

atlas › athena › Merge requests › !31570

Merged Created 1 year ago by **Christos Anastopoulos** Developer Edit

auto_ptr to unique_ptr (5) ATLASRECTS-5296

Overview 3 Commits 2 Pipelines 1 Changes 69

`auto_ptr` to `unique_ptr` ATLASRECTS-5296

Conclusions

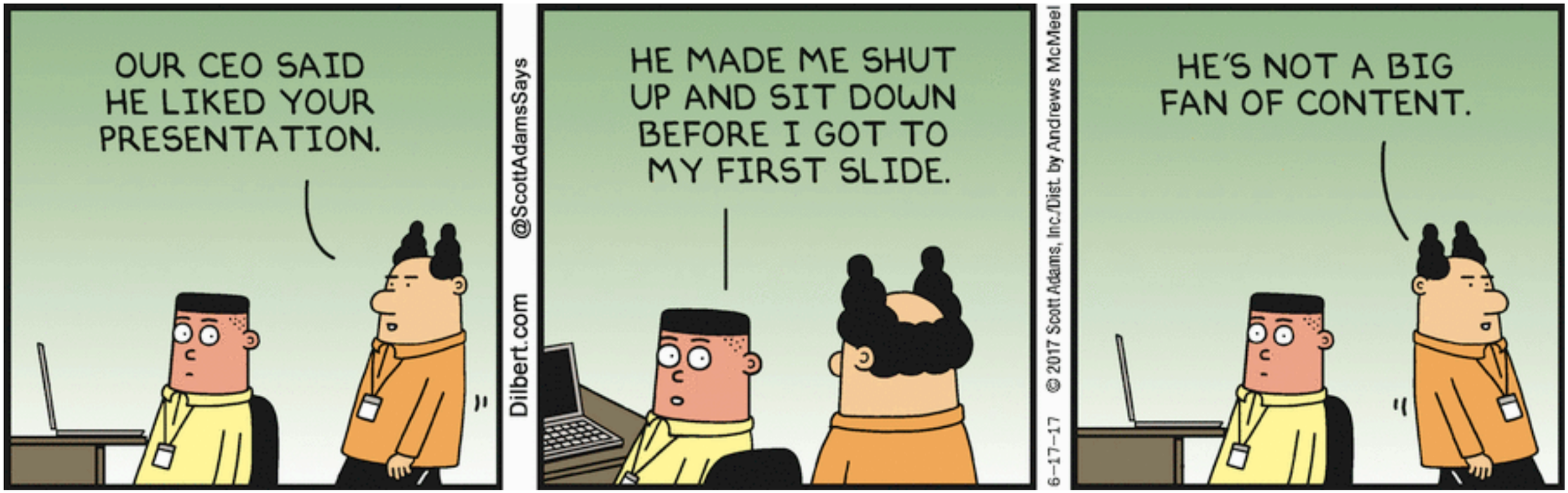
The ATLAS collaboration is at the end of long development cycle for its Run-3 software

This included a significant technical component of migrating to Multi-threading

Personally, not sure I could even contemplate how we would have managed this without rigorous development and testing procedures.

For the Physics you will need to keep an eye for upcoming Run-3 results.

τέλος


OUR CEO SAID HE LIKED YOUR PRESENTATION.
HE MADE ME SHUT UP AND SIT DOWN BEFORE I GOT TO MY FIRST SLIDE.
HE'S NOT A BIG FAN OF CONTENT.
@ScottAdamsSays
Dilbert.com
6-17-17
© 2017 Scott Adams, Inc./Dist. by Andrews McMeel

Backup

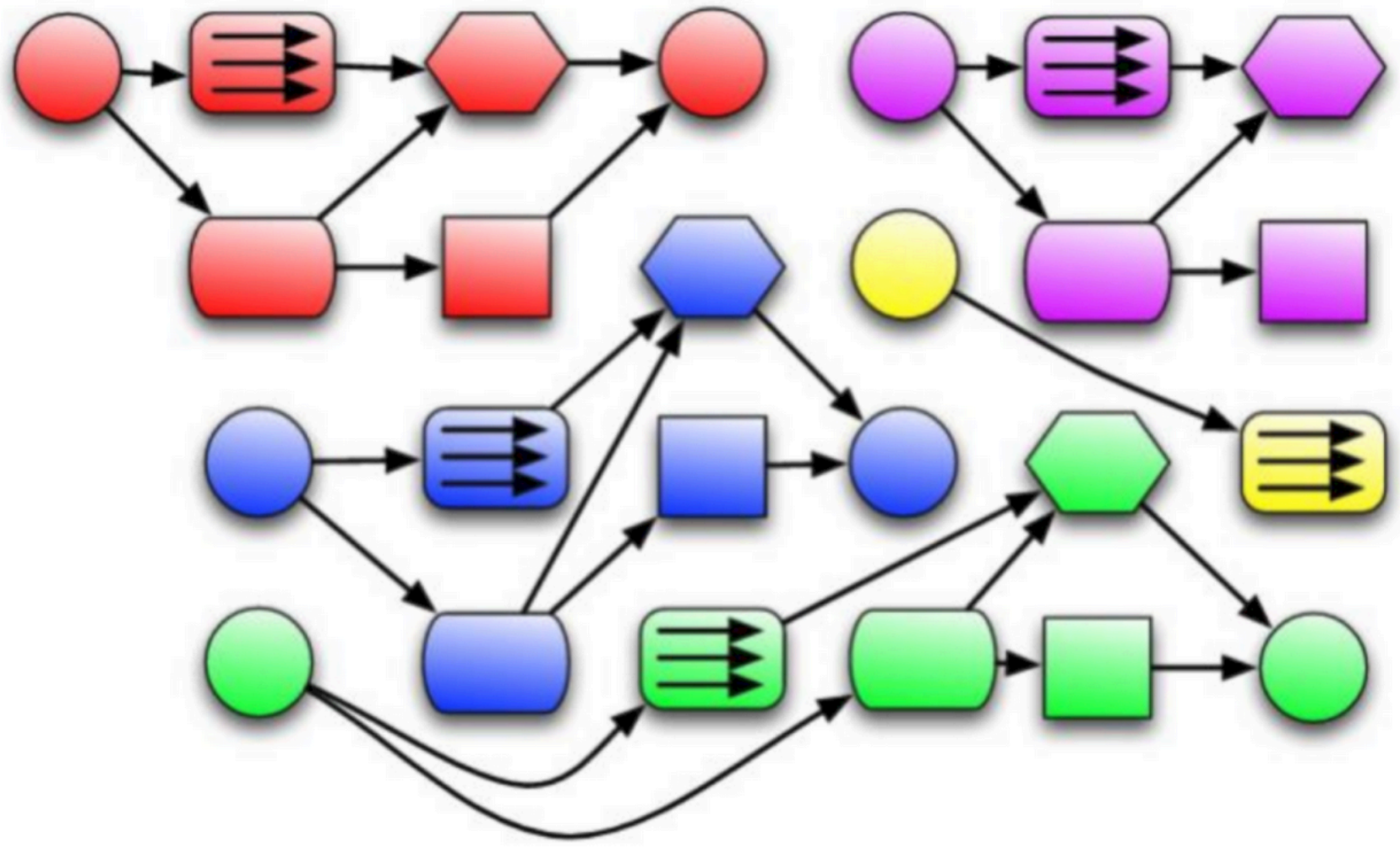
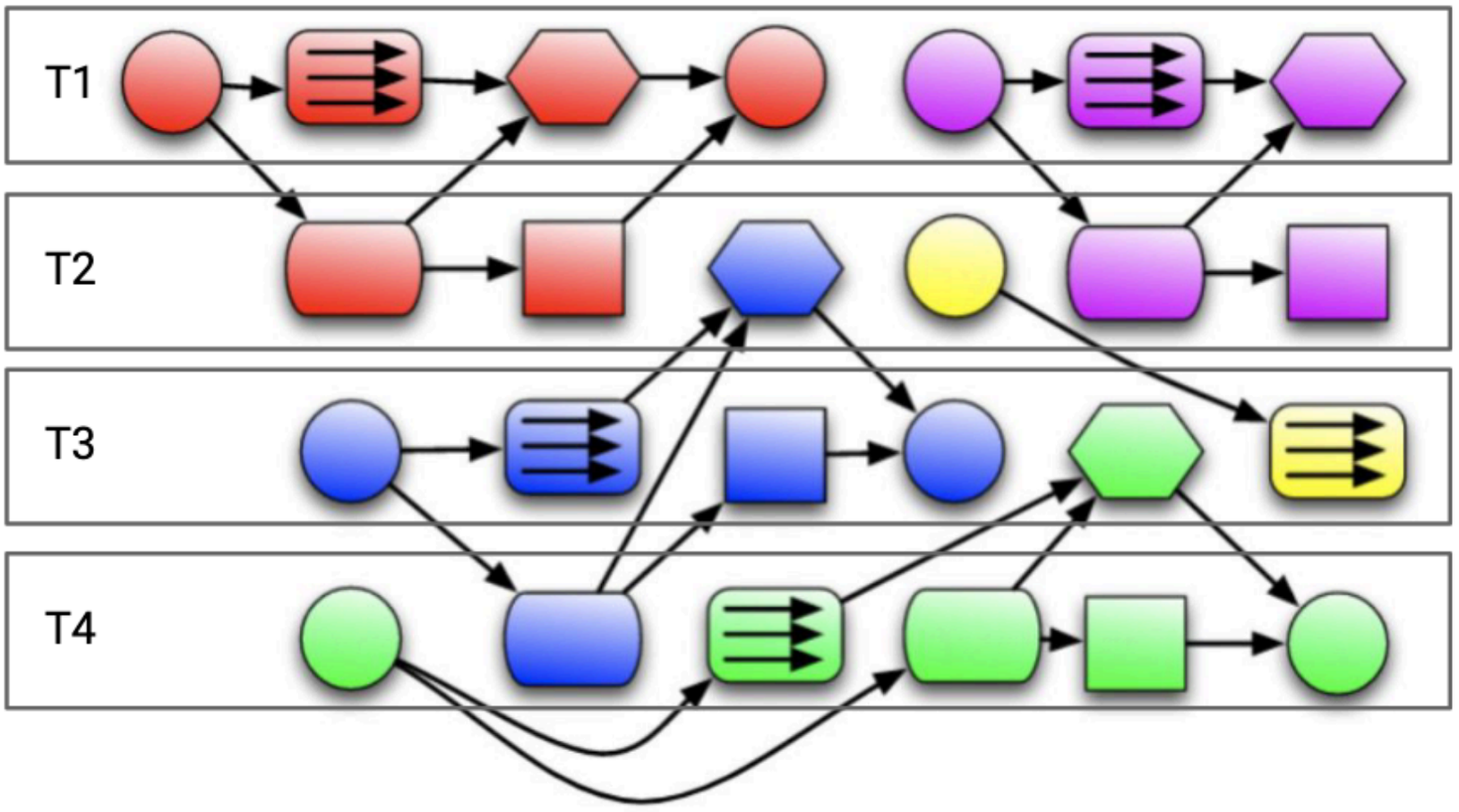
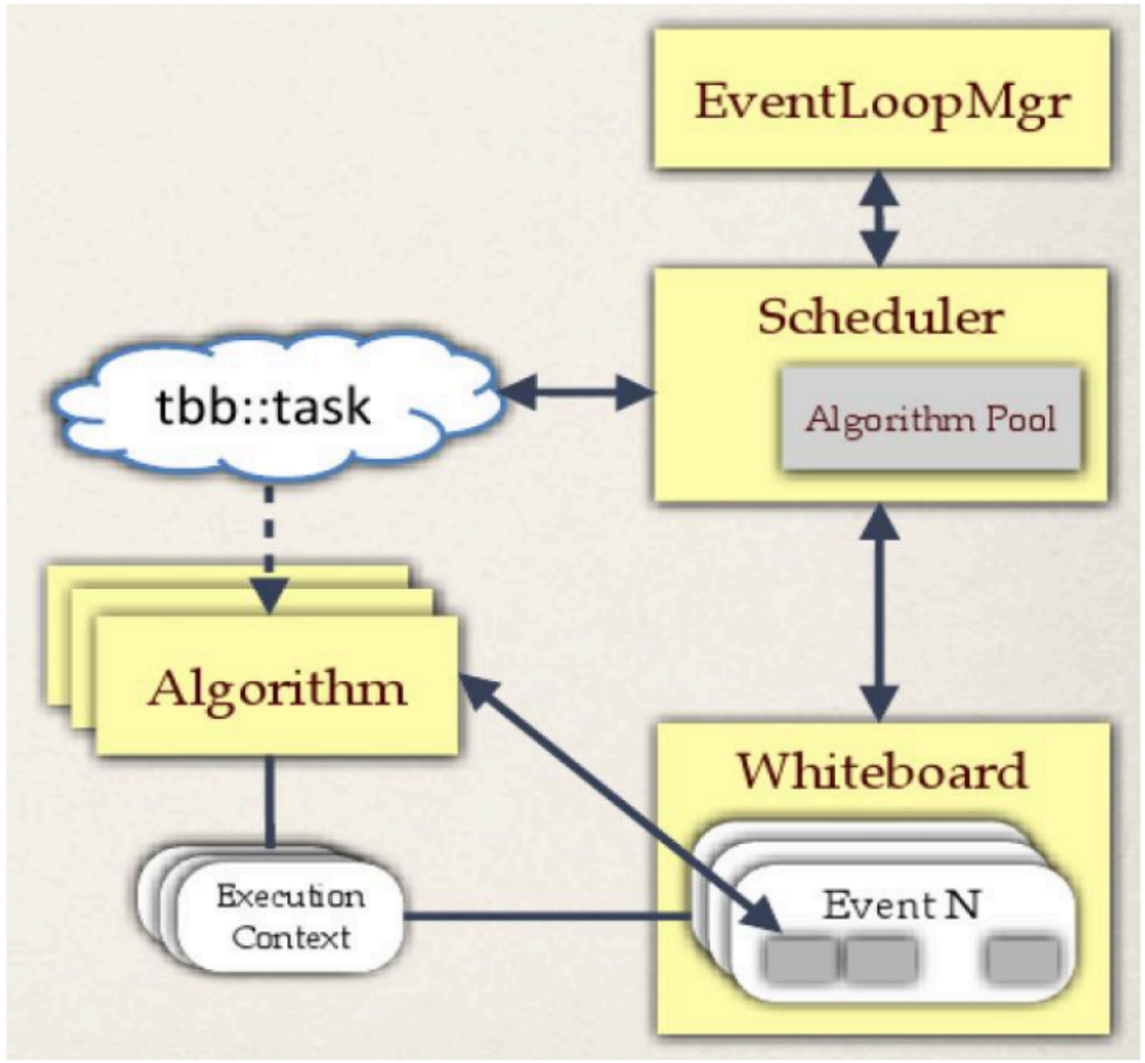


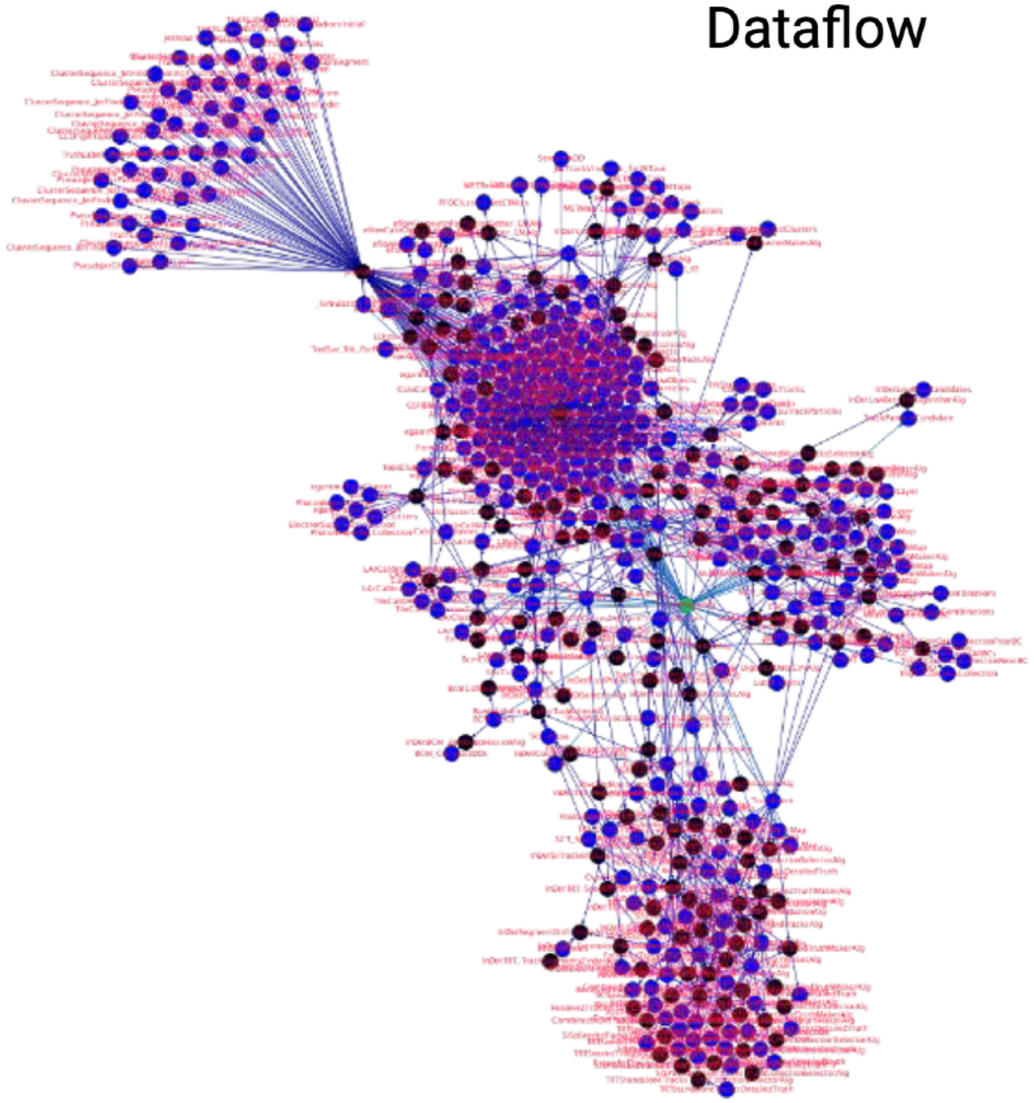
Figure 2: An example of multithreaded execution in AthenaMT. Four threads are shown, each corresponding to one row. Different events are shown with different colours, and different algorithms are shown with different shapes. The algorithms are executed as soon as their input data are available and a thread is free. (Image: ATLAS Collaboration/CERN)



Event processing in AthenaMT:

- Each event is a different color
- Each shape is a different algorithm
- T1-T4: threads #1-4


EventLoopMgr
Scheduler
Algorithm Pool
tbb::task
Algorithm
Whiteboard
Execution
Context
Event N


Dataflow