Future of ROOT
Runtime C++ modules

Yuka Takahashi - Princeton University, CERN
Vasil Geogiev Vasilev - Princeton University

3 Very Important Summary Slides

1. With Runtime C++ Modules,

Experiments will be more Performant

2. With Runtime C++ Modules,

End users don’t have to change anything in their code

3. When you use Runtime C++ Modules,

Set -Druntime_cxxmodules=ON
(Available in future 6.16!)

Agenda

1. C++ Modules in a nutshell
2. Effects for experiments
3. Effect for ROOT
4. Implementation
5. Status and roadmap

C++ Modules in a Nutshell

C++ Modules in a Nutshell

```
#include <vector>
```

C++ Modules in a Nutshell

```
#include <vector>
```

Textual Include	PCH	Modules
✗ Expensive ✗ Fragile	✗ Inseparable	✓

C++ Modules in a Nutshell

```
#include "TVirtualPad.h"
#include <vector>
#include <set>

int main() {
…
```

original code

Preprocess

Textual Include

```
                                    ……
                                    ……
# 286 "/usr/include/c++/v1/vector" 
namespace std { inline namespace __
template <bool> class __vector_base
{
   __attribute__
((__visibility__("hidden"),
__always_inline__)) __vector_base_c
{}
                                    ……
# 394 "/usr/include/c++/v1/set" 3
namespace std {inline namespace __1
template <…> class set {
public:
    typedef _Key key_type;
                                    ……

int main {
                                    ……
```

TVirtualPad.h

mmon

vector

on()

set

one big file!

Parse

Compile

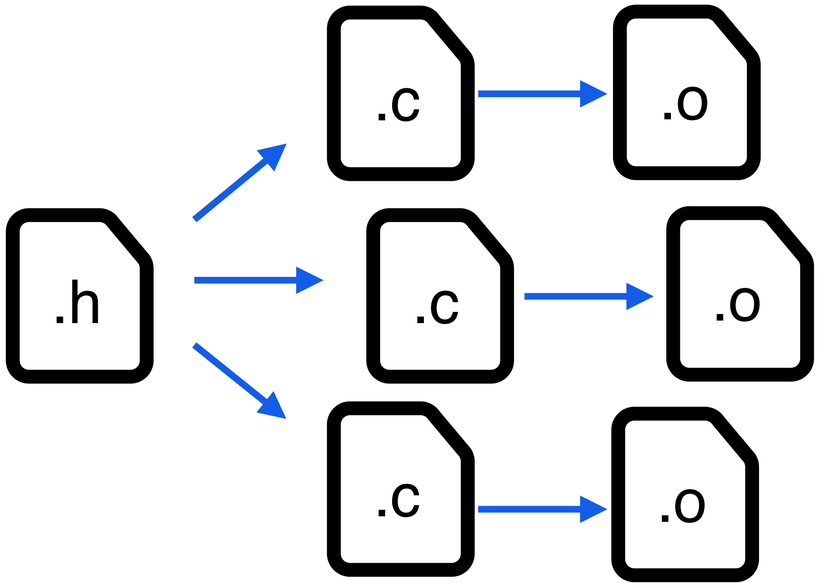
.o

C++ Modules in a Nutshell

Textual Include

1. Expensive
   Reparse the same header

2. Fragile
   Name collisions



Rcpp library

```
#define PI 3.14
…
```

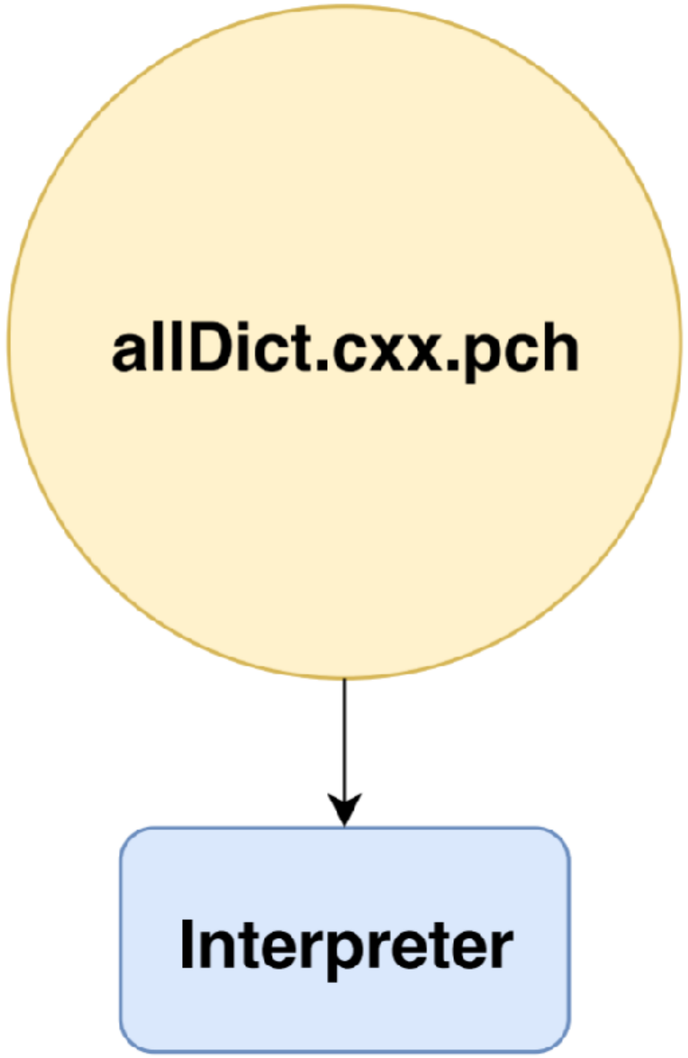
Users' code

```
#include <header.h>
                …
double PI = 3.14;
// => double 3.14 = 3.14;
```

C++ Modules in a Nutshell

PCH (Pre Compiled Header)

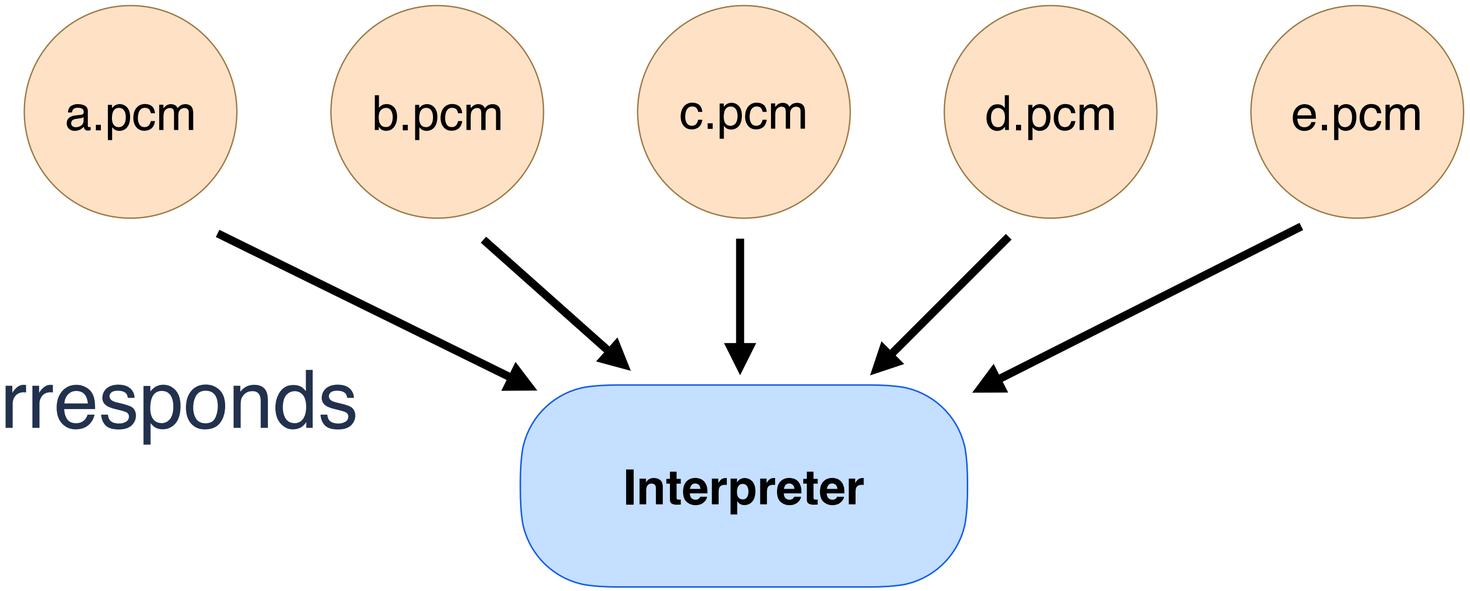
1. Storing pre compiled header information (same as modules)
2. Stored in one big file



C++ Modules in a Nutshell

Modules

- Pre compiled **PCM files** contain header information
- PCMs are **separated**

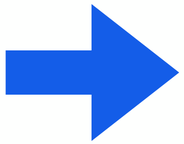


Each PCM file (a.pcm) corresponds to a library (liba.so)

C++ Modules in a Nutshell

Modules

- Pre compiled **PCM files** contain header information
- PCMs are **separated**



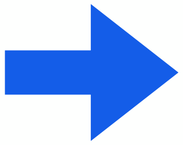
✓Compile-time scalability
✓Fragility
✓Separable

Effect for Experiments

Effect for Experiments

Current problem of Experiment Software Stack

- Experiments are still using textual include
- Not even PCH
- PCH can’t be used because it’s **too big**

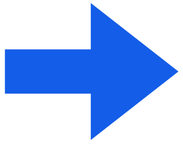


Experiments are parsing **hundreds of headers** at the startup time at the moment

Effect for Experiments

Current problem of Experiment Software Stack

- Experiments are still using textual include
- Not even PCH
- PCH can’t be used because it’s **too big**



Modules can do this as it’s **separable**

Effect for Experiments

Current Status

Compile ROOT in CMS environment with modules ✓

Generate dictionary for CMS external libraries WIP

Effect for Experiments

Current Status

Working closely with CMSSW

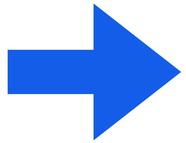
Getting a lot of feedback from CMS usage

- Leads to fix bugs in Clang & Cling

Effect for Experiments

Performance Benefits

No actual benchmark yet

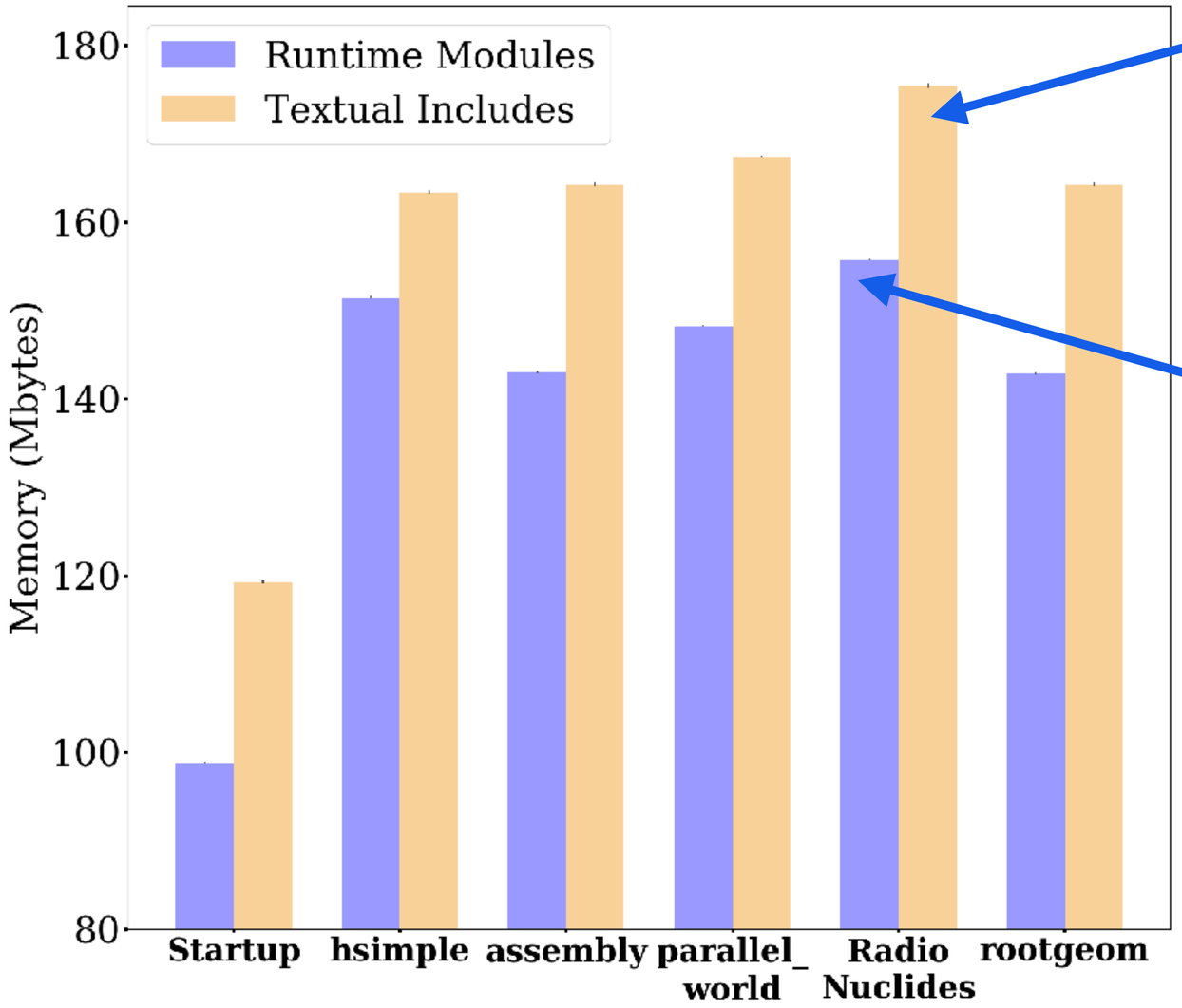


Simulated results in next slides

Comparison of modules to textual includes

Effect for Experiments

Memory

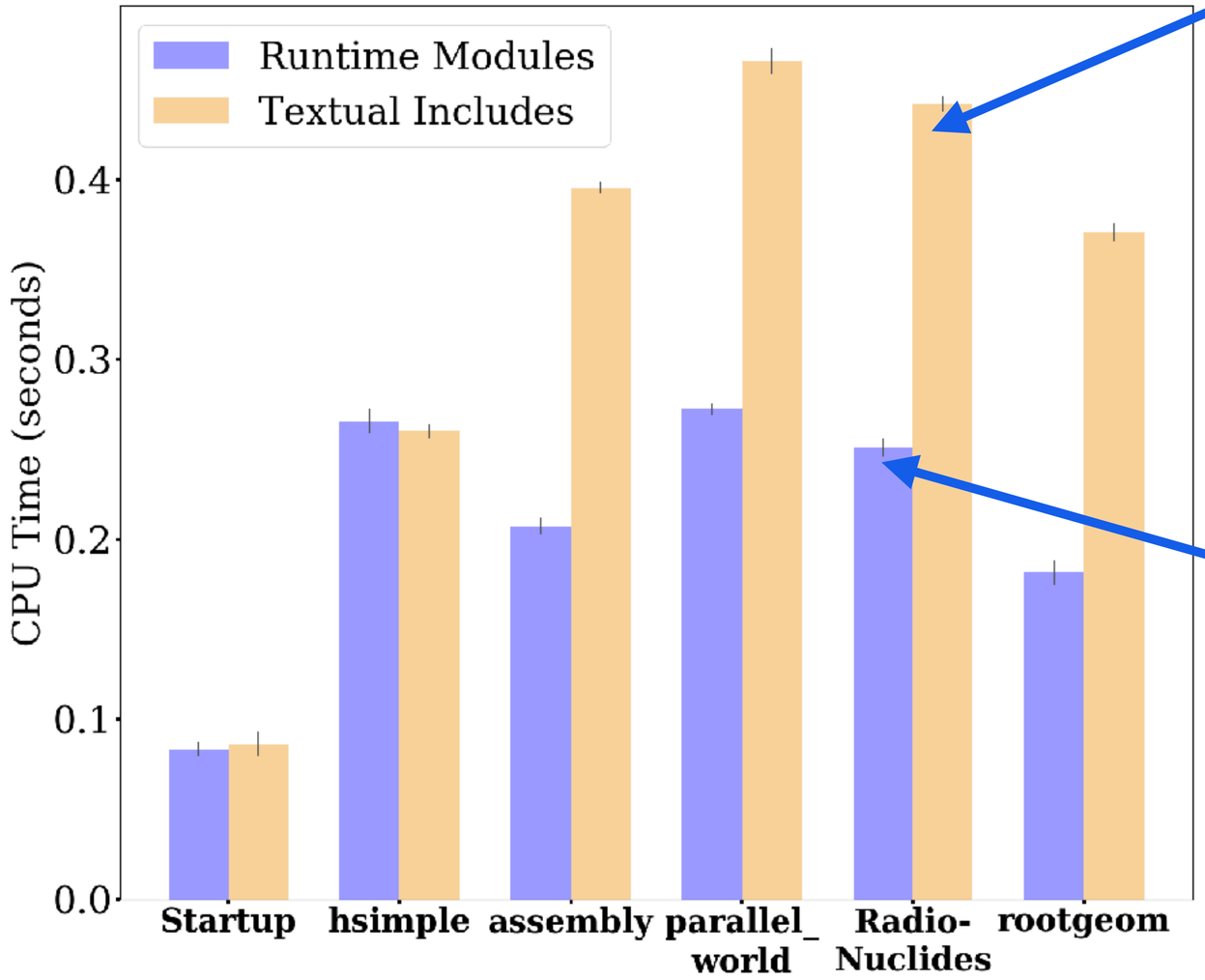


Textual Include

Modules

Effect for Experiments

CPU Time

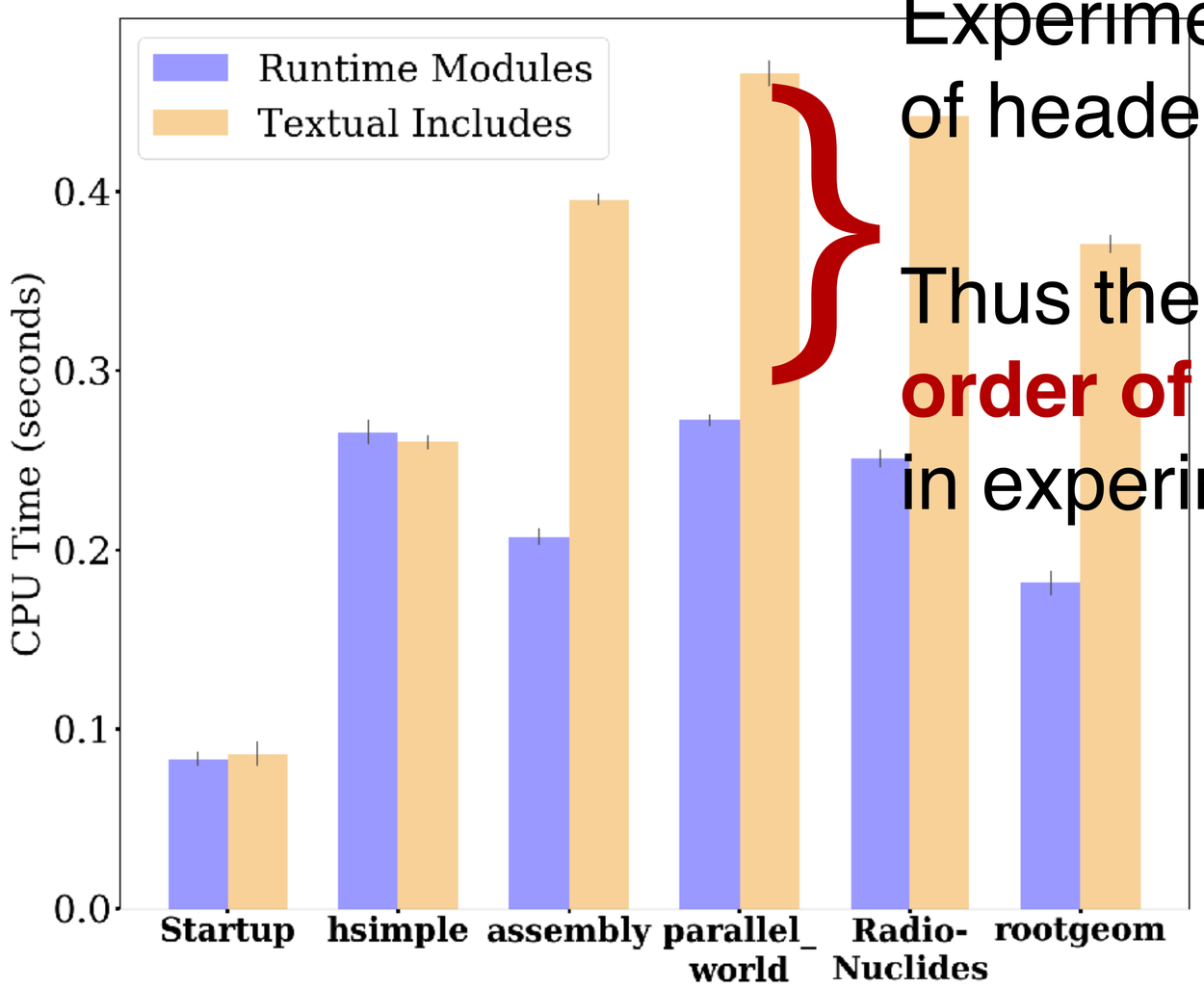


Textual Include

Modules

Effect for Experiments

CPU Time



Experiments have hundreds of header files to parse

Thus the difference will be **order of magnitude** larger in experiments

Effect for ROOT

Effect for ROOT

Correctness benefit

Without Modules

```
$ root -l
root [0] gMinuit // Cannot load variable
IncrementalExecutor::executeFunction:
symbol 'gMinuit' unresolved while
linking [cling interface function]!
```

Effect for ROOT

Correctness benefit

With Modules

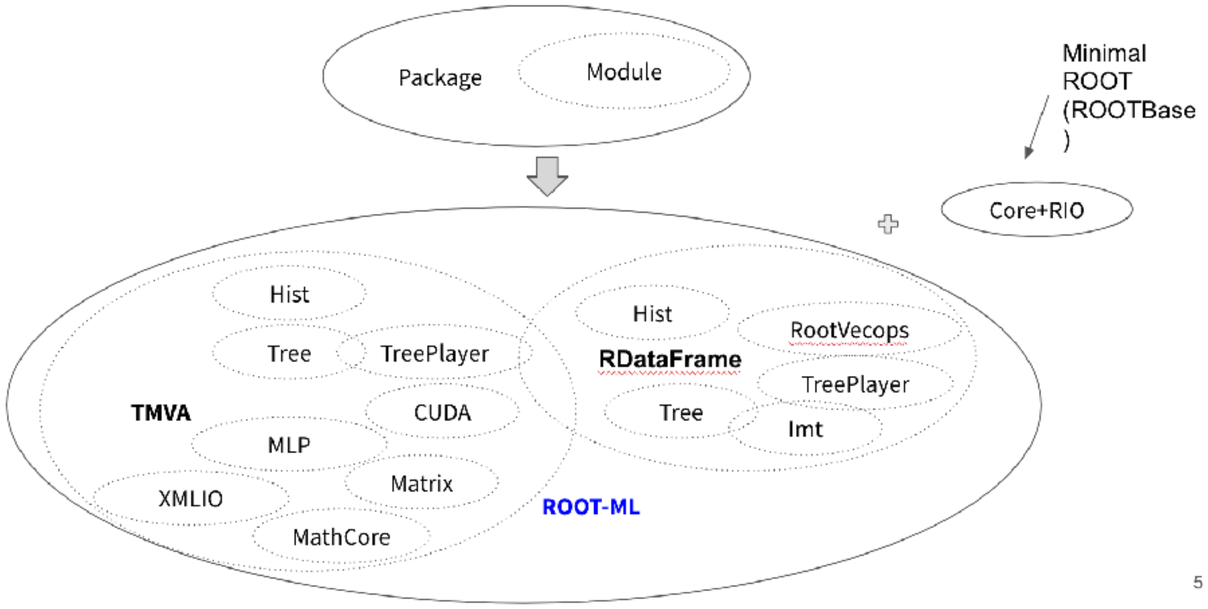
```
$ root -l
root [0] gMinuit // Could load libMinuit
(TMinuit *) nullptr
```

Effect for ROOT

Packaging benefit

We can make ROOT modular for lazy installing packages

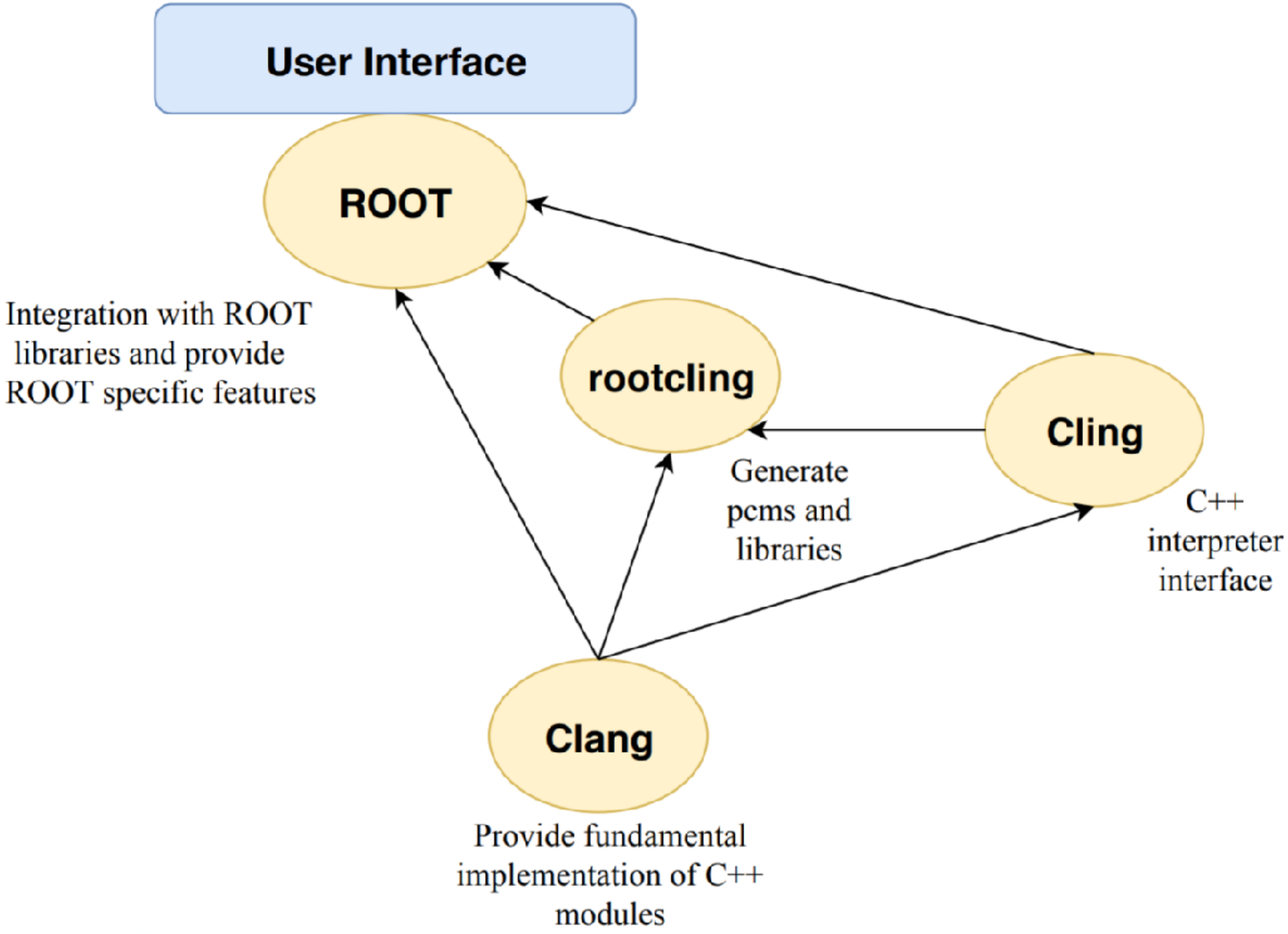
- ROOT package manager
- See Oksana's talk for more information!



Slide from Oksana :)

Implementation

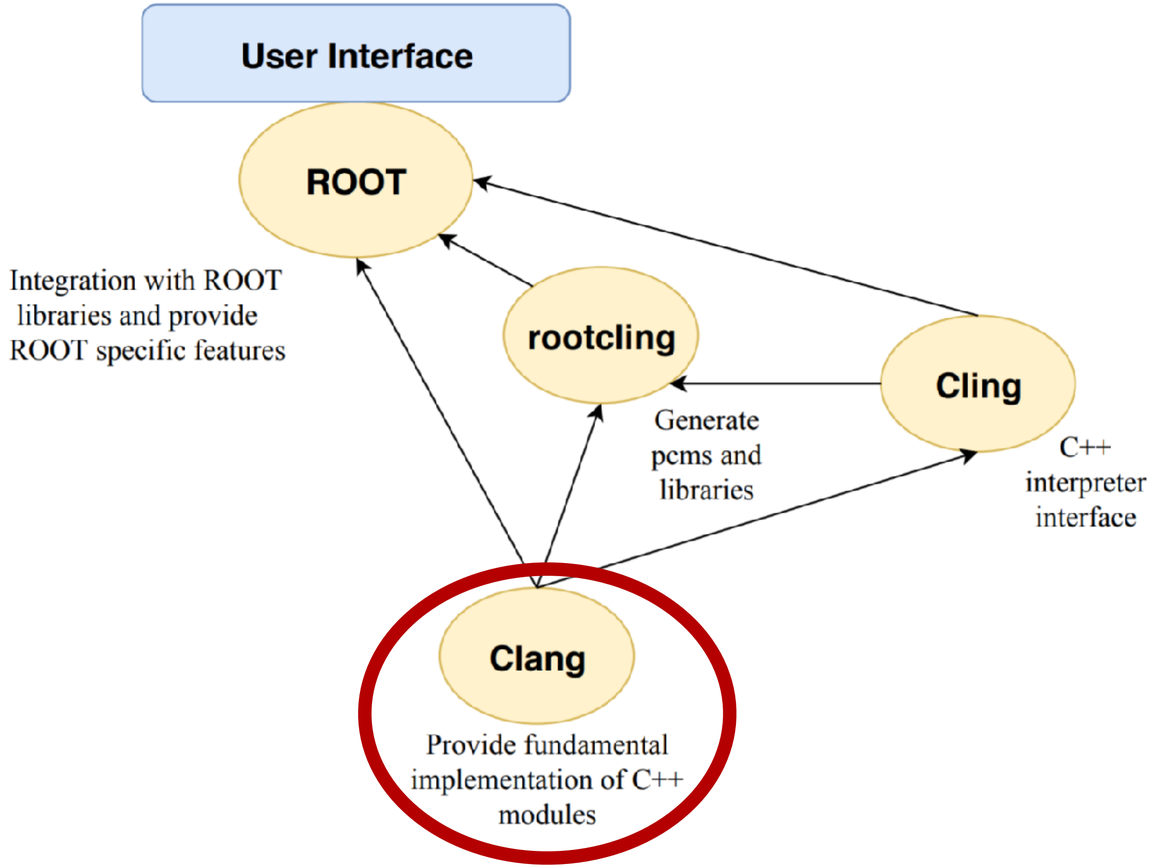
Implementation



Implementation

Clang

- External project under LLVM
- Bi-weekly meeting with C++ Modules community
- Reporting & fixing bugs
- ROOT is the largest user of Modules outside industry



Status and roadmap

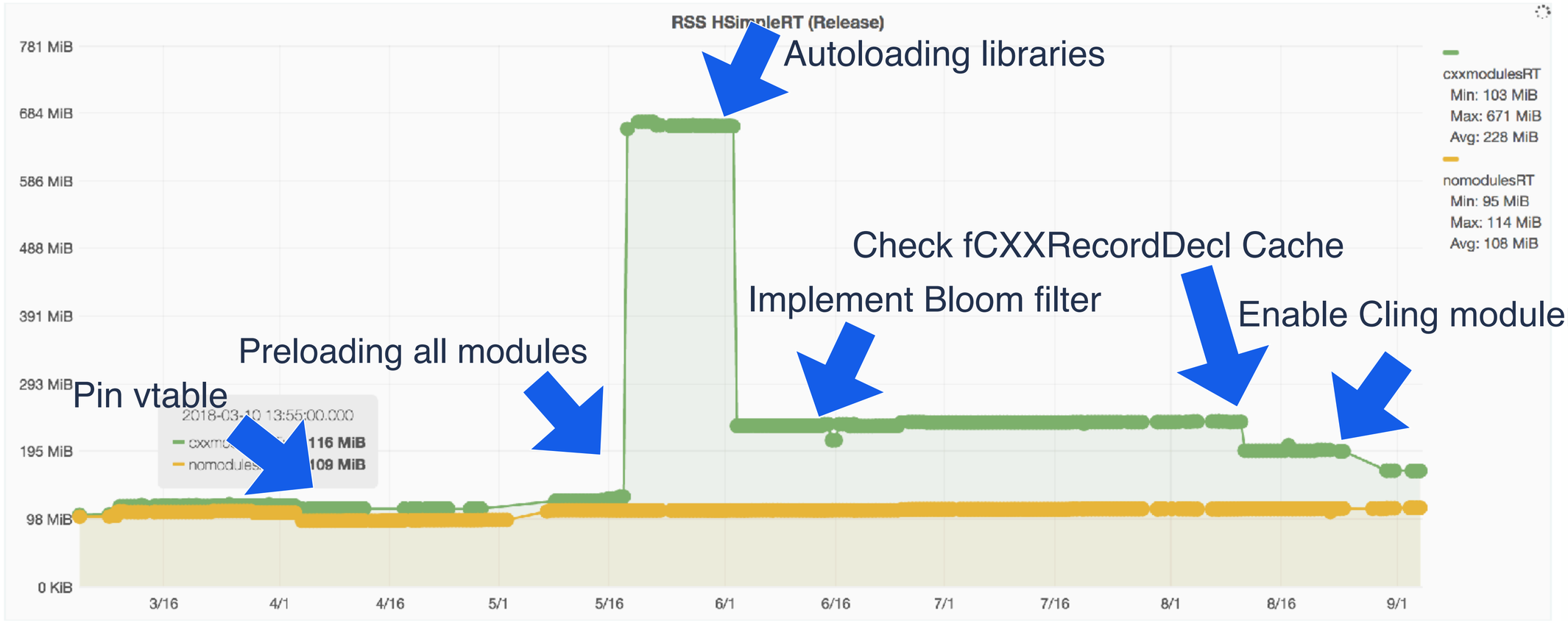
Status and roadmap

Memory - hSimple

Yellow line is PCH

Green line is Modules

https://rootbnch-grafana-test.cern.ch/



…. And More!!

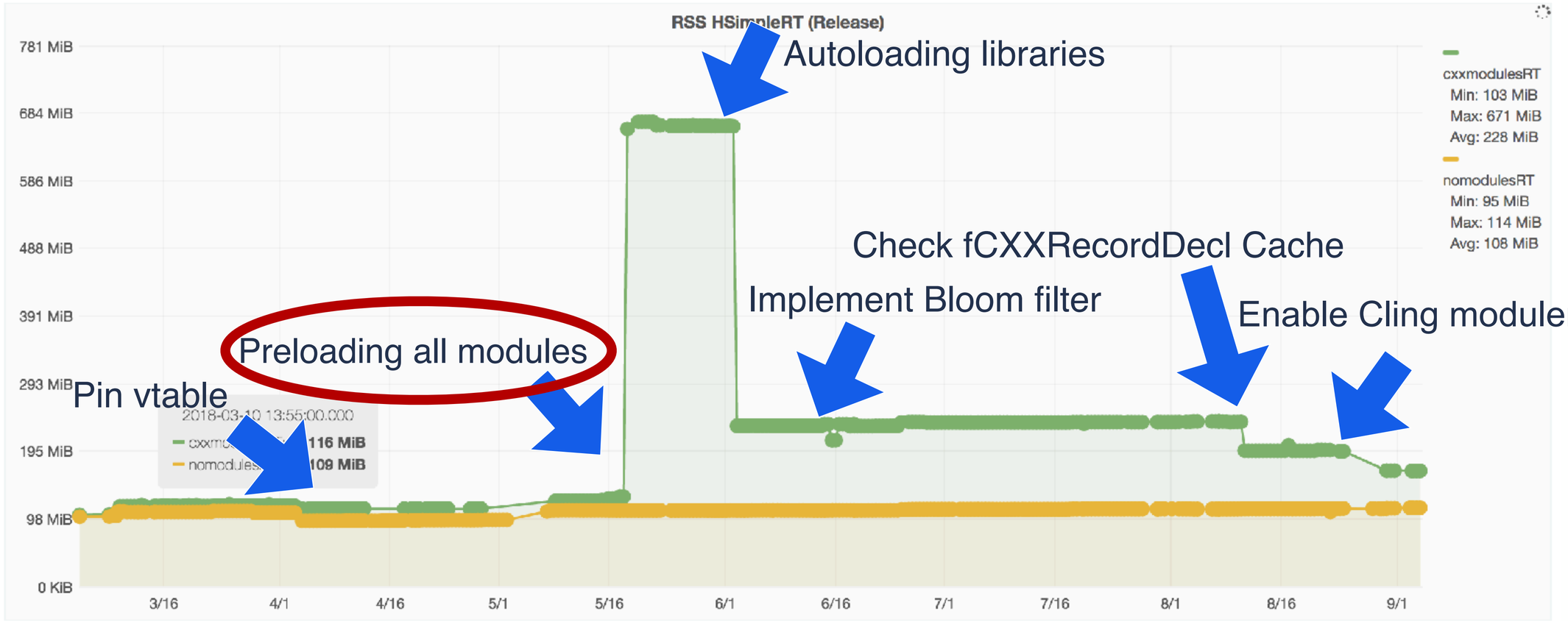
Status and roadmap

Memory - hSimple

Yellow line is PCH

Green line is Modules

https://rootbnch-grafana-test.cern.ch/



…. And More!!

Status and roadmap

Preloading Modules

All correctness benefit over PCH is due to this

- Preloading of all modules
- Replace old infrastructure
  - rootmap

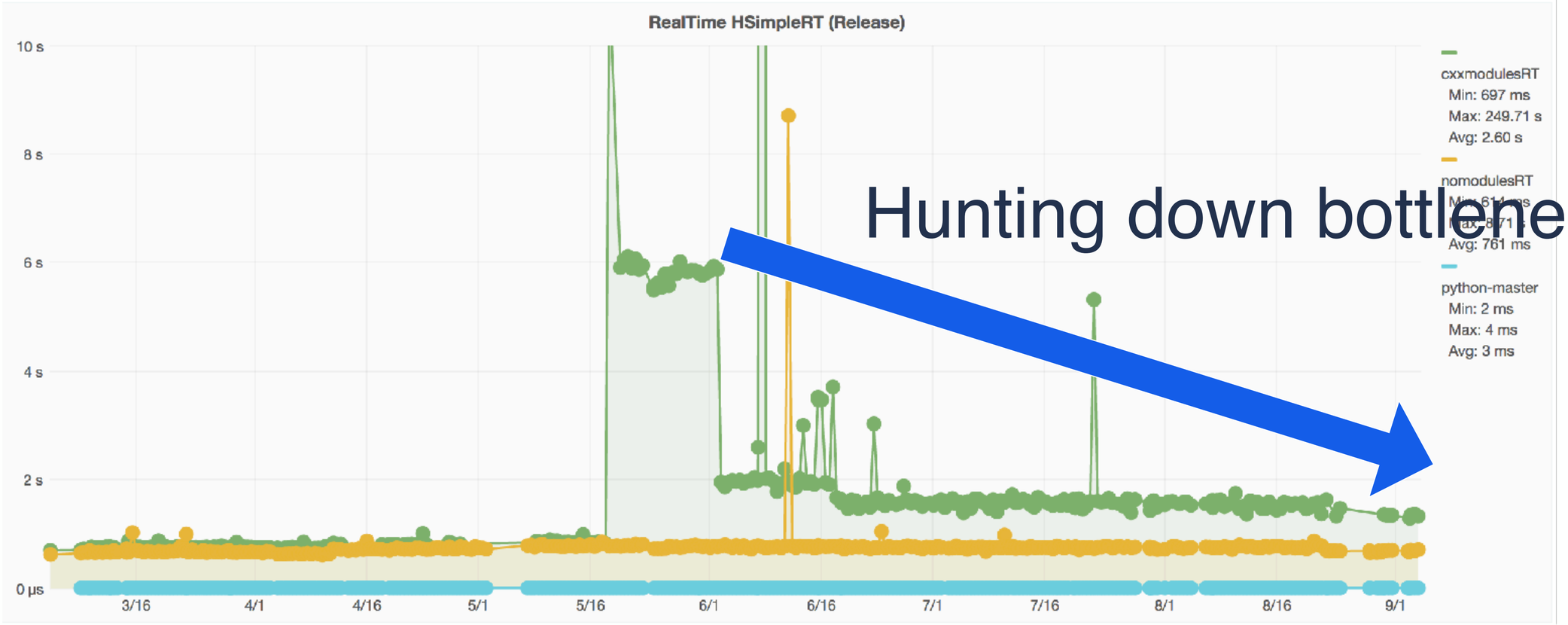
Status and roadmap

Real time - hSimple

Yellow line is PCH

Green line is Modules

https://rootbnch-grafana-test.cern.ch/



Status and roadmap

Status

- Fundamental Construction in ROOT Core, which affects every code passed to ROOT
- Working with industry and CMSSW
- Good progress in performance optimization

Roadmap

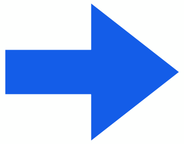
- Reach complete production level before 6.16
- Continue working on optimization
- Modularise CMSSW!

Thank you for your attention!

Backup Slides

Effect for ROOT

C++ Modules is a mechanism to boost **compilation time**



For ROOT, it turns into **runtime speed improvement** as we have C++ interpreter