Templates & STL

Stefan Roiser, Lorenzo Moneta
CERN
PH/SFT

Current Usage

- Template types are highly used in HEP
  - Via STL
  - Libraries provided by external parties
    - E.g. Boost
  - Libraries provided by CERN developers
    - E.g. SMatrix, a library for linear algebra
      - Uses expression templates

Symbols

- Problems
  - Length of symbols
    - Source code type

```
typedef std::map<std::string, std::pair<int,int> > MyMap;
```

    - Mangled symbol (constructor)

```
__ZNSt3mapISsSt4pairIiiESt4lessISsESaIS0_IKSsS1_EEEC1Ev
```

    - Demangled symbol

```
std::map<std::basic_string<char, std::char_traits<char>, std::allocator<char> >,
std::pair<int, int>, std::less<std::basic_string<char, std::char_traits<char>, std::
allocator<char> > >, std::allocator<std::pair<std::basic_string<char, std::char_
traits<char>, std::allocator<char> > const, std::pair<int, int> > > >::map()
```

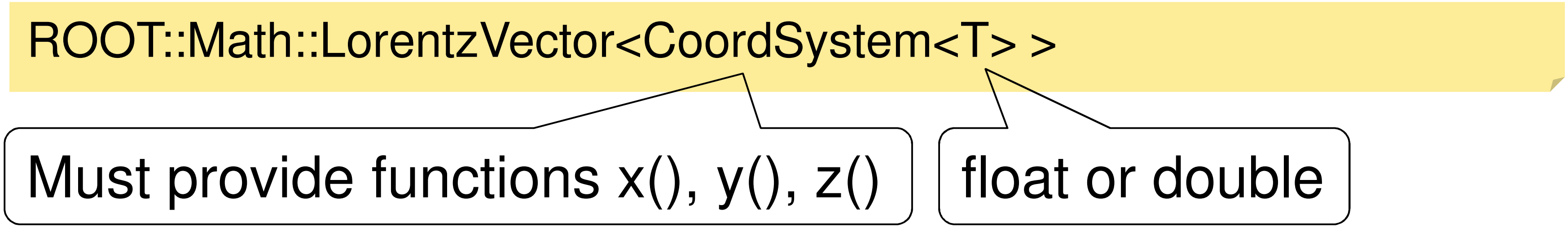
  - Number of symbols
    - Symbols for member functions of class<T> for every instantiation
- Results in
  - Code bloat

Generic vector<T*> ?

    - Many of these symbols are not even used
  - Increased library sizes and load time
    - E.g. literal type names are stored in reflection libraries
  - Lengthy compiler error messages

Concepts

- Narrowing template types a la “concepts”



```
ROOT::Math::LorentzVector<CoordSystem<T> >
```

Must provide functions x(), y(), z()

float or double

- Currently a wrong instantiation produces

```
LorentzVector.h: In member function
'typename CoordSystem::Scalar LorentzVector<CoordSystem>::x()
const [with CoordSystem = Foo]':
testMessage.cxx:13: instantiated from here LorentzVector.h:616:
error: 'const struct Foo' has no member named 'Px'
```

Other Issues

- Thread safety for STL?
  - E.g. “Intel Threading Building Blocks”
    - Provides implementations for hash_map, queue, vector
- Enhancements for iterator programming
  - Simplifications for the coding itself
    - E.g. using lambda functions