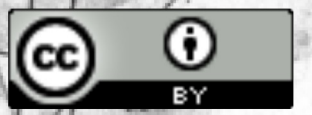
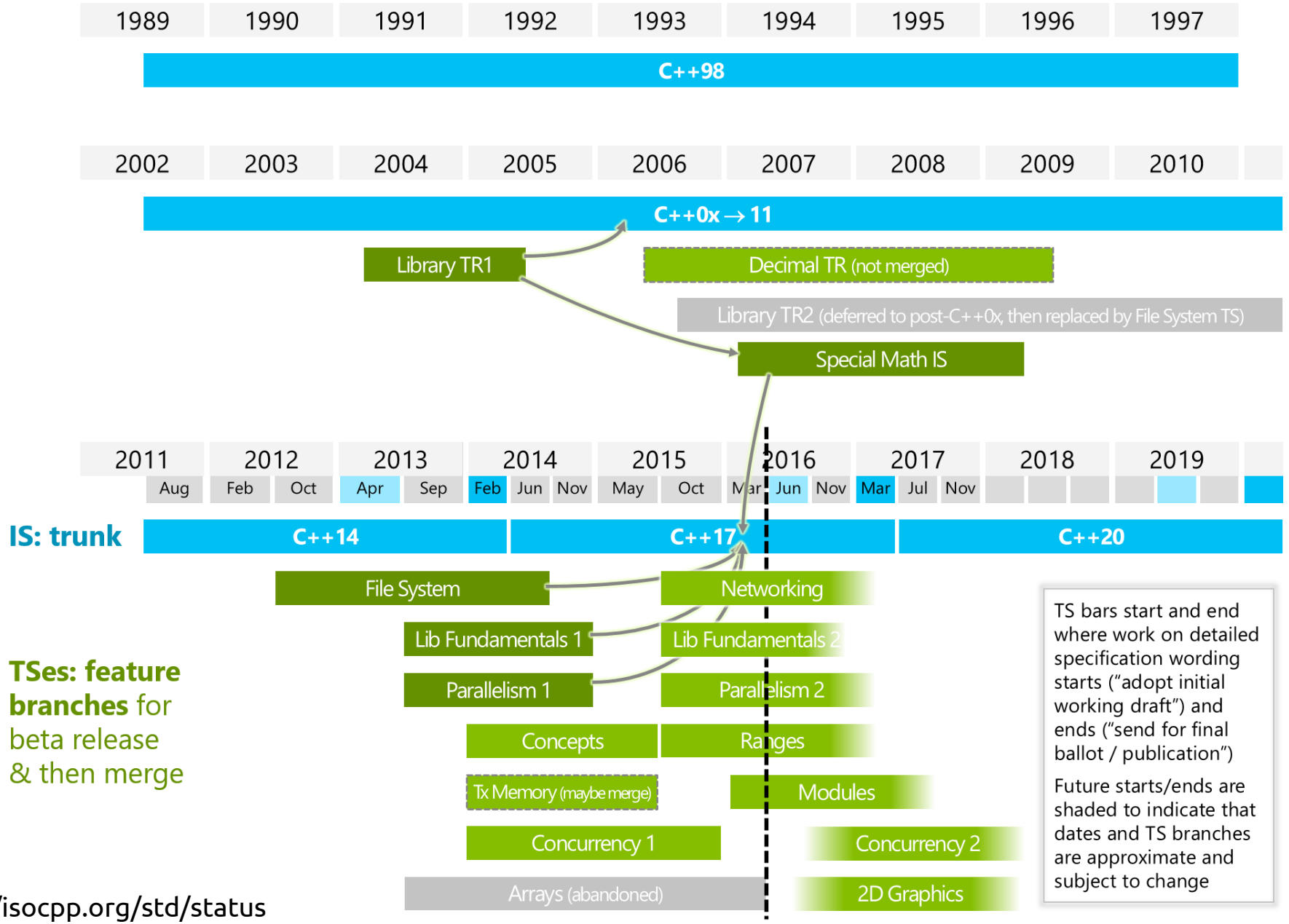
C++ Static analysis Sanitizers

Francesco Giacomini
INFN-CNAF

Workshop CCR
La Biodola, 16-20 maggio 2016



C++ Status



https://isocpp.org/std/status

Static analysis

- C++ è un linguaggio *statically-typed*
  - il compilatore è in grado di verificare che le espressioni in un programma siano corrette dal punto di vista dei tipi usati
  - il compilatore si limita a verificare che un programma sia *well-formed*
- La tecnologia alla base dei compilatori può essere tuttavia utilizzata per fare ulteriori analisi di qualità e correttezza
  - tecnologia largamente disponibile grazie all'esistenza di diversi compilatori, open-source e proprietari
- Alcuni tool disponibili su https://ci.infn.it/
  - istanza Jenkins nazionale
  - inclusi in immagini Docker

OCLint

http://oclint.org/

"Static code analysis tool for improving quality and reducing defects by inspecting C, C++ and Objective-C code and looking for potential problems like: [...]"

Total Files	Files with Violations	Priority 1	Priority 2	Priority 3	Compiler Errors	Compiler Warnings	Clang Static Analyzer
13	11	0	23	25	0	7	1

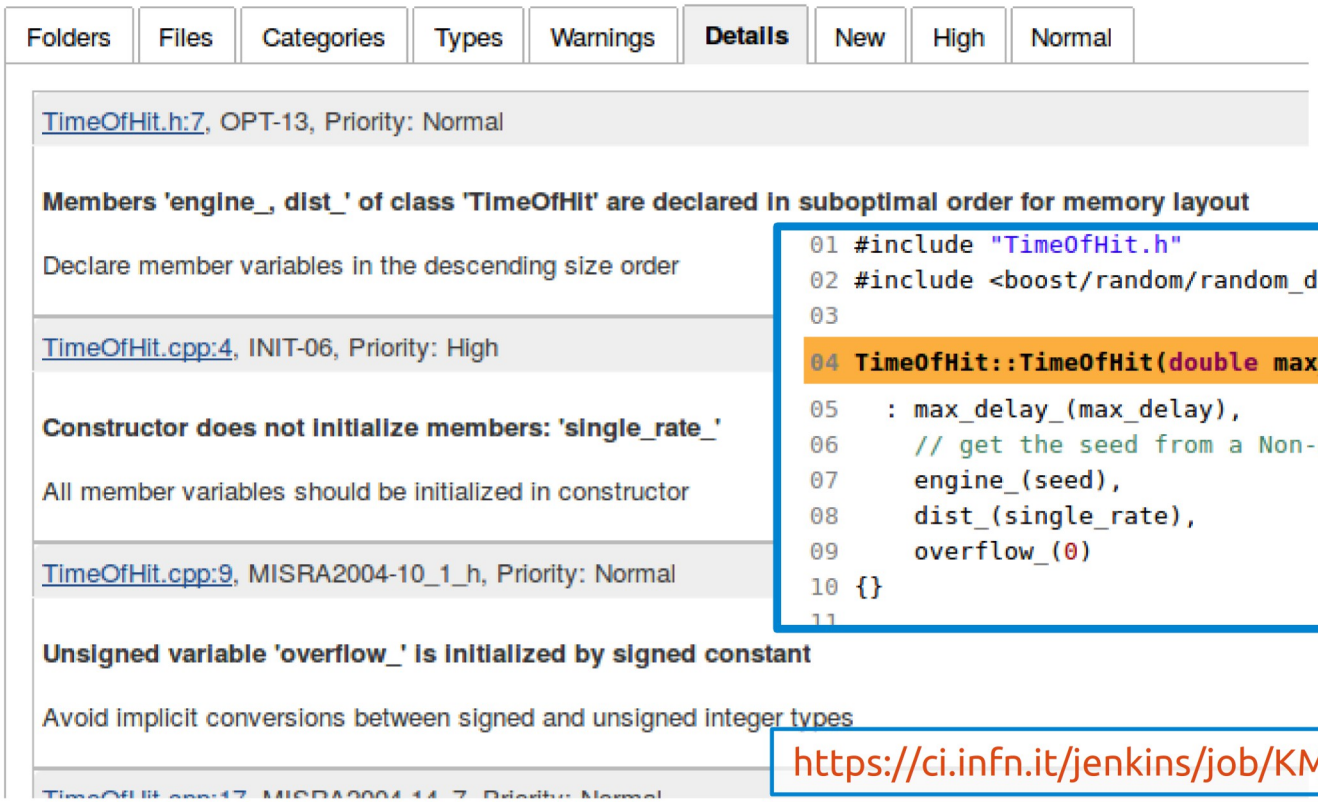
File	Location	Rule Name	Rule Category	Priority	Message
/home/giaco/Downloads/tridas /DAQ/TSC/tsc.cpp	35:1	long method	size	3	Method with 100 lines exceeds limit of 50
/home/giaco/Downloads/tridas /DAQ/TSC/tsc.cpp	35:1	high cyclomatic complexity	size	2	Cyclomatic Complexity Number 13 exceeds limit of 10
/home/giaco/Downloads/tridas /DAQ/TSC/tsc_tests.cpp	85:1	clang static analyzer		checker bug	Address of stack memory associated with local variable 'dev_null' is still referred to by the global variable 'sm' upon

https://ci.infn.it/jenkins/job/KM3/job/tridas_static-analysis/288/
(include output di cppcheck e cloc, OCLint in formato PMD)

Parasoft C/C++test

https://www.parasoft.com/product/cpptest/

"Integrated Development Testing solution for automating a broad range of testing best practices proven to improve development team productivity and software quality"




```
01 #include "TimeOfHit.h"
02 #include <boost/random/random_device.hpp>
03
04 TimeOfHit::TimeOfHit(double max_delay, double single_rate, unsigned int seed)
05   : max_delay_(max_delay),
06     // get the seed from a Non-deterministic Uniform Random Number Generator
07     engine_(seed),
08     dist_(single_rate),
09     overflow_(0)
10 {}
```

https://ci.infn.it/jenkins/job/KM3/job/tridas_cpptest/1/CpptestResult/

Sanitizers

- Run-time analysis
- Nel codice binario vengono inseriti controlli di validità e correttezza, valutati durante l'esecuzione
  - overhead
- Address Sanitizer, `-fsanitize=address`
  - https://github.com/google/sanitizers/wiki/AddressSanitizer
  - Use after free, Heap buffer overflow, Stack buffer overflow, Global buffer overflow, Use after return, Initialization order bugs, Memory leaks
- Undefined Behavior Sanitizer, `-fsanitize=undefined`
  - http://clang.llvm.org/docs/UndefinedBehaviorSanitizer.html
  - Using misaligned or null pointer, Signed integer overflow, Conversion to, from, or between floating-point types which would overflow the destination, ...
  - L'ottimizzazione sfrutta *undefined behaviour* in modi sorprendenti
- ...

UBSan

```
$ DAQ/TSC/RunTSC -n

/home/giaco/Downloads/boost-1.58/include/boost/container/scoped_allocator_fwd.hpp:83:24: runtime error:
reference binding to null pointer of type 'const struct allocator_arg_t'

/home/giaco/Downloads/boost-1.58/include/boost/container/scoped_allocator_fwd.hpp:83:24: runtime error:
reference binding to null pointer of type 'const struct allocator_arg_t'

/home/giaco/Downloads/boost-1.58/include/boost/lexical_cast/detail/converter_lexical_streams.hpp:236:43: runtime
error: downcast of address 0x7ffc2a97ce38 which does not point to an object of type 'basic_unlockedbuf'

0x7ffc2a97ce38: note: object is of type 'std::basic_stringbuf<char, std::char_traits<char>, std::allocator<char>
>'

 16 7f 00 00  48 9c b2 89 16 7f 00 00  49 48 79 02 00 00 00 00  49 48 79 02 00 00 00 00  49 48 79 02

              ^~~~~~~~~~~~~~~~~~~~~~

              vptr for 'std::basic_stringbuf<char, std::char_traits<char>, std::allocator<char> >'

/home/giaco/Downloads/boost-1.58/include/boost/lexical_cast/detail/converter_lexical_streams.hpp:236:43: runtime
error: downcast of address 0x7ffc2a97b6c8 which does not point to an object of type 'basic_unlockedbuf'

0x7ffc2a97b6c8: note: object is of type 'std::basic_stringbuf<char, std::char_traits<char>, std::allocator<char>
>'

 16 7f 00 00  48 9c b2 89 16 7f 00 00  19 bc 7a 02 00 00 00 00  19 bc 7a 02 00 00 00 00  19 bc 7a 02

              ^~~~~~~~~~~~~~~~~~~~~~

              vptr for 'std::basic_stringbuf<char, std::char_traits<char>, std::allocator<char> >'
```

Uso in UQ

- Uncertainty Quantification - Holistic quality quantification in Monte Carlo simulation

Physics Validation

- Photoelectric cross sections
  - Published April 2016
- Electron backscattering
  - 2 publications 2015, 3rd being written
- e^+e^- pair production by photons
- Electron impact ionisation cross sections
  - 3 students at Milano-Bicocca involved
- Photonuclear interactions
  - Starting, collaboration with ENEA

Software Quality

Large scale production of software quality metrics over Geant4 0.0-10.2 version with new version of Imagix4D

- Analysis in progress
- Innovative statistical analysis incorporating econometric methods

Uncertainty Quantification

- Paper on mathematical methods under review

M. G. Pia, E. Ronchieri