Document Number: P1445R0
Date: 2019-01-21
Authors: Michael Wong
Project: Programming Language C++, SG1
Reply to: Michael Wong <michael@codeplay.com>

Concurrency TS: to update or not update

Contents

Discussion

Concurrency TS was published as N4577 (based on p0159r0) in 2016-01-19. It is ISO/IEC TS 19571:2016. It contains three major parts:

- Futures
- Atomic smart pointers
- Latches and barriers

Since then we have changed significant portions, including moving latches and barriers in P1135, and adapting atomic<shared_ptr<T>> into IS20. Futures was not moved as much there is a wish to replace it with a Facebook-style dependency handling, which will likely come after C++ 20. This will likely enter Concurrency TS2.

However, in SAN a defect fix was generated for LWG 2697:
http://www.open-std.org/jtc1/sc22/wg21/docs/papers/2018/p1224r0.html#2697
It is about unwrapping constructor when given an invalid `future`

But given that future in the TS is not moving in C++20, and likely to be completely replaced by the Facebook effort, this paper aims to motivate discussion on whether we should republish N4577 just to fix this defect, given that significant part of the futures aspect will be completely replaced.

Finally, the remaining third part of the original concurrency TS (other than futures, latches and barrier), atomic smart pointer has already entered the IS20 now as atomic<shared_ptr<T>>. That essentially takes care of all parts of this TS.

So as far as updating the Concurrency TS for the defect, I am not sure we should. This is the point that needs discussion.