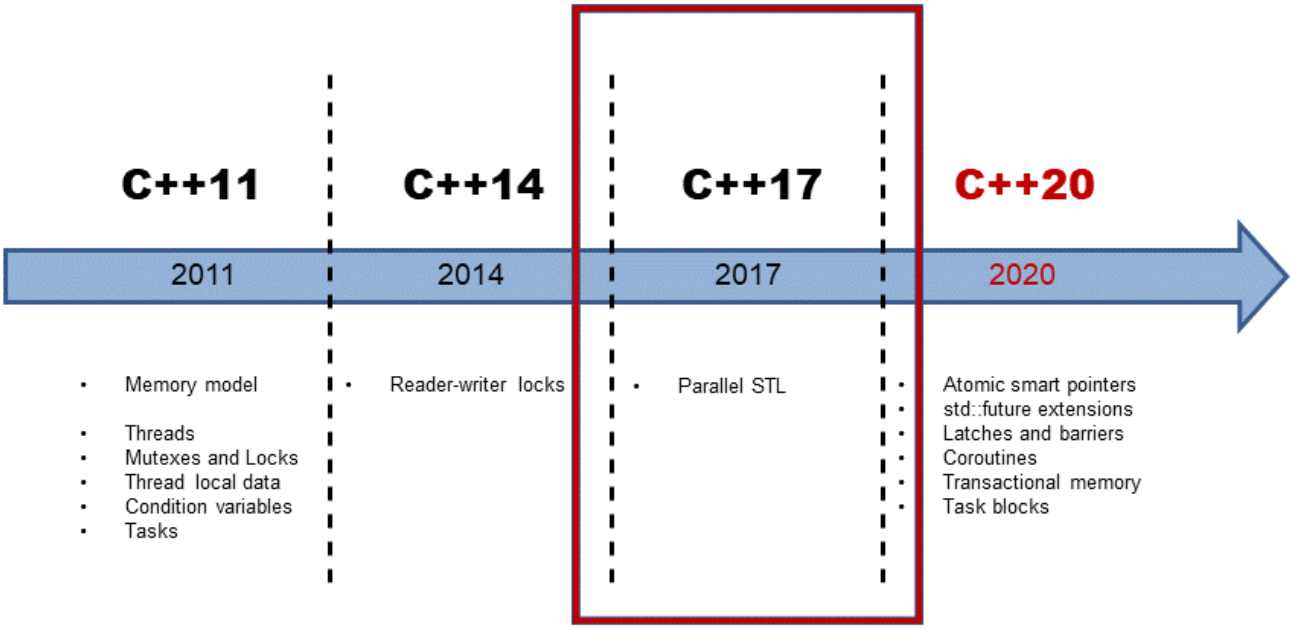
C++17: Parallel Algorithms of the Standard Template Library

A short introduction of parallel algorithms of the standard template library in C++17.

WE'LL COVER THE FOLLOWING



With C++17, concurrency in C++ has drastically changed - particularly for the parallel algorithms of the Standard Template Library (STL). C++11 and C++14 only provide the basic building blocks for concurrency. These tools are suitable for a library or framework developer, but not for the application developer. Multithreading in C++11 and C++14 will become an assembly language for concurrency in C++17!

Execution Policy

With C++17, most of the STL algorithms will be available in a parallel implementation. This makes it possible for you to invoke an algorithm with a so-called execution policy. This policy specifies whether the algorithm runs sequentially `std::seq`, in parallel `std::par`, or in parallel with additional vectorization `std::par_unseq`.

New Algorithms

In addition to the 69 algorithms that are available for parallel or vectorized executions in overloaded versions, we get eight new algorithms. These new ones are well suited for parallel reducing, scanning, or transforming.