Compiler Support

Let's look at the compiler support for the topics in this chapter.

As of today (July 2019) only two compilers/STL implementation support parallel algorithms: it's Visual Studio (since 2017 15.7) and GCC (since 9.1).

Visual Studio implements `par_unseq` as `par`, so you shouldn't expect any difference between code runs.

GCC implementation uses modified Intel PSTL and relies on OpenMP 4.0 and Intel TBB 2018. You need to install and link with `-ltbb` if you want to work with parallel algorithms.

For example:

```
g++-9.1 -std=c++17 -Wall -O2 sample.cpp -ltbb
```

For Building GCC 9.1 and Intel TBB you can check this guide @Solarian Programmer: C++17 STL Parallel Algorithms - with GCC 9.1 and Intel TBB on Linux and macOS.

Summary:

Feature	GCC	Clang	MSVC
Parallel Algorithms	9.1[^pargcc]	in progress	VS 2017 15.7[^parstlmsvc]

> *[^pargcc]:* See in the article: GCC 9.1 Released and GCC 9 Release Series — Changes, New Features, and Fixes) *[^parstlmsvc]:* See Announcing: MSVC Conforms to the C++ Standard | Visual C++ Team Blog

There are also several other implementations out there:

- Codeplay - SYCL Parallel STL
- STE||AR Group - HPX
- Intel - Parallel STL - based on OpenMP 4.0 and Intel® TBB.
- Parallel STL - early Microsoft implementation for the Technical Specification.
- n3554 - proposal implementation (started by Nvidia)
- Thibaut Lutz Implementation @Github - early implementation

In the next chapter, we will look at other changes in the library.