This project focuses on implementing standard conforming parallel algorithms in HPX. The requirements and specification are defined by open-std standards proposal N4071, a technical specification for C++ extensions for parallelism. HPX is a general purpose C++ runtime system for parallel and distributed applications, and this proposal can benefit HPX greatly with asynchronous algorithms. The new algorithms will introduce a new argument, known as an execution policy. This execution policy object will specify whether the algorithm will run asynchronously or synchronously. All other algorithm requirements are defined by it’s predecessor.

A New Approach

According to standard proposal N4071, an execution policy object can be either sequential, parallel, or vectorized; represented in C++ as seq, par and vec. Our HPX implementation however offers one additional policy, task. A task execution policy will specify whether the algorithm will run asynchronously or synchronously. All other algorithm requirements are defined by it’s predecessor.

```
using hpx::parallel;
//
std::for_each(v.begin(), v.end(), func);

//execute for_each in parallel
hpx::future<void> r = for_each(task, v.begin(), v.end(), func);

//do work

//now synchronize
r.wait();
```